

High Performance Storage System User Guide

Table of Contents

| | |
|---|----|
| Preface | 4 |
| Introduction | 6 |
| Organization | 7 |
| Typographic Conventions | 7 |
| Storage Summarized | 8 |
| 1. Overview | 9 |
| 1.1. User Interfaces | 10 |
| 1.1.1. File Transfer Protocol (FTP) | 10 |
| 1.1.2. Parallel FTP (PFTP) | 10 |
| 1.1.3. Network File System Version 2 (NFS V2) | 11 |
| 1.1.4. IBM SP Parallel I/O File System (PIOFS) Import/Export | 12 |
| 1.1.5. NFT (LC Local Interface) | 12 |
| 1.1.6. User Utilities | 13 |
| 1.2. Storage Concepts | 14 |
| 1.2.1. Class of Service | 14 |
| 1.2.2. Storage Class | 15 |
| 1.2.3. Storage Hierarchy | 15 |
| 1.3. Interface Usage Considerations | 17 |
| 1.4. User IDs | 19 |
| 1.5. DCE User Accounts | 20 |
| 2. File Transfer Protocol (FTP) | 21 |
| 2.1. Site Commands | 23 |
| 2.1.1. SETCOS (Specifying a File's Class of Service) | 23 |
| 2.1.2. CHGID (Changing a File's Group by ID) | 25 |
| 2.1.3. CHGRP (Changing a File's Group by Name) | 25 |
| 2.1.4. CHMOD (Changing a File's Permissions) | 26 |
| 2.1.5. CHOWN (Changing a File's Owner by Name) | 27 |
| 2.1.6. CHUID (Changing a File's Owner by ID) | 27 |
| 2.1.7. STAGE (Staging a File) | 27 |
| 2.1.8. WAIT (Setting the Desired Wait Options for Migrated Files) | 27 |
| 2.2. List Directory Extensions | 30 |
| 3. Parallel File Transfer Protocol (PFTP) | 31 |
| 3.1. Additional Commands | 32 |
| 3.1.1. PAPPEND (Parallel append) | 33 |
| 3.1.2. PPUT (Parallel file store) | 34 |
| 3.1.3. MPPUT (Parallel file store) | 36 |
| 3.1.4. PGET (Parallel file retrieval) | 37 |
| 3.1.5. MPGET (Parallel file retrieval) | 39 |
| 3.1.6. PSOCKET (Specify TCP socket based transfers) | 40 |
| 3.1.7. PIP3 (Specify IPI-3 based transfers) | 41 |
| 3.1.8. SETPWIDTH (Specify transfer stripe width) | 42 |
| 3.1.9. SETPBLOCKSIZE (Specify transfer block size) | 44 |
| 3.1.10. PARALLEL (LLNL only) | 46 |

| | |
|---|----|
| 3.1.11. HTAR (PFTP Front End, LLNL only) | 47 |
| 4. IBM SP PIOFS Import/Export | 48 |
| 4.1. Command Interface | 48 |
| 4.1.1. PIOFSIE (Import or export a file between PIOFS and HPSS) | 48 |
| 4.1.2. Import/Export Parameter File | 52 |
| 4.1.3. Configuration Variables and Setup | 54 |
| 5. User Utilities | 56 |
| 5.1. Utilities | 56 |
| 5.1.1. CHACL (Change ACL utility) | 56 |
| 5.1.2. LSACL (List ACL utility) | 59 |
| 5.1.3. LSHPSS (List information about HPSS) | 61 |
| 6. Storage Groups | 64 |
| Appendix A. Acronyms | 67 |
| Appendix B. References | 68 |
| Appendix C. HPSS/NSL UniTree Differences | 69 |
| C.1. Trash Cans Discontinued | 69 |
| C.2. Supplementary Commands Changed | 69 |
| C.3. Listing Options Changed | 70 |
| Disclaimer | 71 |
| Keyword Index | 72 |
| Alphabetical List of Keywords | 74 |
| Date and Revisions | 76 |

Preface

Scope: This High Performance Storage System (HPSS) User's Guide provides the necessary information for transferring files to or from LLNL's installation of HPSS. This reference manual is adapted with permission from similar publications whose copyright owners are listed in the notes below, to reflect HPSS use at Livermore Computing, LLNL.

Local enhancements of and restrictions on file storage are included in appropriate places throughout the text (especially in the PFTP section, where the LLNL PFTP client has features not available at other sites).

LC users seeking a concise, illustrated introduction to local storage features, limitations, and customized storage-support tools should consult the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide, rather than trying to locate highly relevant material in this much larger standard HPSS manual.

Availability: Unless otherwise noted, HPSS and its features are available on both the open and the secure computing facilities at LLNL.

Consultant: For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: lc-hotline@llnl.gov, SCF e-mail: lc-hotline@pop.llnl.gov).

Note: Copyright (C) 1996, 1997 International Business Machines Corporation
Copyright (C) 1996, 1997 Lockheed Martin Energy Research Corporation and Oak Ridge National Laboratory (ORNL)
Copyright (C) 1996, 1997 Regents of The University of California and Lawrence Livermore National Laboratory (LLNL)
Copyright (C) 1996, 1997 Regents of The University of California and Los Alamos National Laboratory (LANL)
Copyright (C) 1996, 1997 Sandia Corporation and Sandia National Laboratories (SNL)
Copyright (C) 1996, 1997 NASA Langley Research Center (LaRC)
All rights reserved
Printed in the United States of America
HPSS Release 3 August 1997 (Revision 2)

IBM is a registered trademark of International Business Machines Corporation. AIX and RISC/6000 are trademarks of International Business Machines Corporation. Encina is a registered trademark of Transarc Corporation. UNIX is a registered trademark of Unix System Laboratories, Inc. Sammi is a trademark of Scientific Software Intercomp. NFS and Network File System are trademarks of Sun Microsystems, Inc. DST is a trademark of Ampex Systems Corporation. ACLS is a trademark of Storage Technology Corporation. Other brands and product names appearing herein may be trademarks or registered trademarks of third parties.

Printing: The print file for this document can be found at:

on the OCF: <http://www.llnl.gov/LCdocs/hpss/hpss.pdf>

on the SCF: https://lc.llnl.gov/LCdocs/hpss/hpss_scf.pdf

Introduction

This document provides the User's Guide information for HPSS Release 3.2. In particular, the following interfaces are described: standard File Transfer Protocol (FTP) interface, Parallel FTP (PFTP) interface, Network File System Version 2 (NFS V2), and IBM SP Parallel I/O File System (PIOFS) Import/Export. Note: It is not the intent of this document to define the standard commands and subcommands provided by standard FTP and NFS. Only interface extensions provided by HPSS are defined within the HPSS User's Guide. Note also that HPSS has no interface to support SCP (secure copy), so that you cannot use SCP to store files or retrieve stored files at LLNL. HTAR, which at LC acts as a specialized HPSS interface for storing and retrieving archive (TAR-like library) files, uses PFTP behind the scenes. Because "secure FTP" (SFTP) relies on a different server than standard FTP, you cannot use SFTP to store files or retrieve stored files at LLNL.

Refer to the HPSS Administration Guide for a description of the interfaces provided to HPSS administrators. Programming interfaces are documented in the HPSS Programmer's Reference Guides.

Refer to the HPSS Programmer's Reference Guide, Volume 1 for programming interfaces provided to the end user. The programming interfaces provided in the Programmer's Guide Volume 1 are: HPSS Client Application Programming Interface (API), PIOFS / Export, and 64 bit arithmetic functions.

Refer to the HPSS Error Messages Manual for a list of all HPSS error and advisory messages which are out put by the HPSS software. For each message, the following information is provided: message identifier and text, source file name(s) which generated the message, problem description, system action, and administrator action.

Refer to the HPSS Programmer's Reference Guide, Volume 2 for programming interfaces to each HPSS server. While it is envisioned that most users will access HPSS through the client API, standard FTP, PFTP, NFS, or the PIOFS Import / Export interfaces (documented in Volume 1), well defined programming interfaces are defined for each HPSS server. It should be noted that programming to the individual server level is a more complex programming model which requires a greater understanding of the HPSS servers.

LC users seeking a concise, illustrated introduction to local storage features, limitations, and customized storage-support tools should consult the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide, rather than trying to locate highly relevant material in this much larger standard HPSS manual. Standard FTP features as well as local customizations to support parallel transfers to and from storage are explained thoroughly in LC's FTP Reference Manual. (URL: <http://www.llnl.gov/LCdocs/ftp>)

Organization

The HPSS User's Guide is structured as follows:

- Chapter 1: Overview. (page 9) This chapter provides an overview of each type of user interface, a summary of key storage concepts, and recommendations on usage. The standard interfaces not supported at LC, and the local interfaces only available on LC machines are both noted in the text.
- Chapter 2: File Transfer Protocol (FTP). (page 21) This chapter defines the extensions to the standard FTP interface.
- Chapter 3: Parallel File Transfer Protocol (PFTP). (page 31) This chapter defines the Parallel FTP (PFTP) interface (on LC production machines, FTP defaults to PFTP, the local PFTP client has extra features also covered in this chapter, and the HTAR utility runs PFTP behind the scenes).
- Chapter 4: IBM SP PIOFS Import/Export. (page 48) This chapter defines the IBM SP Parallel I/O File System Import/Export interface (not available at LC).
- Chapter 5: User Utilities. (page 56) This chapter defines the set of HPSS utilities available to the general user (at LC, available only to system administrators).
- Appendix A: Acronyms. (page 67) This appendix A provides a list of acronyms used in this document.
- Appendix B: References. (page 68) This appendix lists documents cited in the text as well as other reference materials.

Typographic Conventions

This document uses the following typographic conventions:

- Italic* Italic words or characters represent variable values to be supplied.
- [...] Brackets enclose optional items in syntax and format descriptions.
- {...} Braces enclose a list of items to select in syntax and format descriptions.

Storage Summarized

This section briefly summarizes the chief storage-system constraints, and tells how to perform the most important file-storage tasks at LC. We suggest you save it for ready reference.

Storage System Constraints:

```

Largest allowed file size: 512 Gbyte (using FTP/NFT interface)
                           8 Gbyte/member      | (using HTAR
                           no limit/archive      | interface)
Longest file name:         255 characters
Problem characters in file names:
  Treated as file filters:  ? * {a,b}
  Forbidden FIRST characters: - ! ~
  Forbidden in any position: ' " : ; { } , /

```

Commands for Common File-Storage Tasks:

| TASK: | <u>FTP</u> | <u>NFT</u> |
|---|--------------------------------|---------------------|
| Connect to storage: | ftp storage | nft |
| Make storage directory: | mkdir <i>dr</i> | (same) |
| Change storage directories: | cd <i>dr</i> | (same) |
| Store a file: | put <i>fl</i> | (same) |
| Retrieve a stored file: | get <i>fl</i> | (same) |
| Delete a stored file: | delete <i>fl</i> | (same) |
| List stored files: | dir | (same) |
| Change permissions (to <i>nnn</i>): | quote site chmod <i>nnn fl</i> | chmod <i>nnn fl</i> |
| Start migration of a stored file back from tape: | quote site stage <i>fl</i> | (none) |
| Control file overwriting: | | |
|Prevent overwriting | (none) | noclobber |
|Allow overwriting | (default) | clobber |

HPSS has no interface to support SCP (secure copy) or SFTP (secure FTP), so that you cannot use SCP or SFTP to store files or retrieve stored files at LLNL. Also, LC's firewall now prevents direct FTP connections to storage from any machine outside llnl.gov (see the [FTP section](#) (page 21) below for suggested alternatives). See the [EZSTORAGE](http://www.llnl.gov/LCdocs/ezstorage) (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide for a concise, LC-oriented treatment of local storage features and problems (such as recursive deletions or sharing stored files), as well as an introduction to customized local storage-support tools (such as LSTORAGE, CHMODSTG, CHGRPSTG, and HTAR).

1. Overview

The High Performance Storage System (HPSS) provides scalable parallel storage systems for highly parallel computers as well as traditional supercomputers and workstation clusters. Concentrating on meeting the high end of storage system and data management requirements, HPSS is scalable and designed for large storage capacities, and to use network-connected storage devices to transfer data at rates up to multiple gigabytes per second. Listed below are the user interfaces for accessing data from HPSS, and this diagram shows the basic system components:

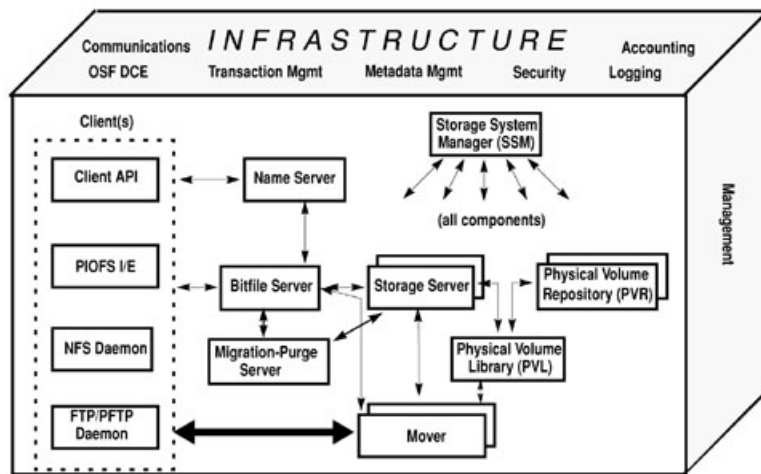


Figure 1-3 The HPSS System

1.1. User Interfaces

1.1.1. File Transfer Protocol (FTP)

HPSS provides an industry-standard FTP user interface. Because FTP is a serial interface, data sent to a user is received serially. This does not mean that the data within HPSS is not stored and retrieved in parallel. It simply means that the FTP daemon within HPSS must consolidate its internal parallel transfers into a serial data transfer to the user. HPSS FTP performance in many cases will be limited not by the speed of a single storage device, as in most other storage systems, but by the speed of the data path between the HPSS FTP daemon and the user's FTP client. [NOTE: on all LC production machines, the default FTP client is really the parallel interface PFTP, and all files over 1 Mbyte move to or from storage in parallel by default.]

All FTP commands are supported or properly rejected if the HPSS Parallel FTP Daemon does not implement a specific feature. In addition, the ability to specify "Class of Service for" is provided via the `quote site` or `site commands`. Additional site command (page 23) options are provided for `chgrp`, `chgid`, `chmod`, `chown`, `chuid`, `stage`, and `wait`. The HPSS FTP Daemon supports access from any RFC-0959 conformant FTP Client.

Different FTP servers process incoming FTP "M" commands (such as MDELETE) differently, with different file-transfer results. The FTP server(s) on HPSS at LC are unusual in interpreting MDELETE filters (such as TEST*) recursively; they remove all (matching) files not only in the current working directory but also in the directory children of that directory as well. To minimize the impact of this aggressive interpretation of MDELETE on your stored files, see the "FTP Pitfalls (with Storage)" section of the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) guide.

Passive connections are not supported. Also, to avoid confusion, the user may want to explicitly specify the data transfer type of `ascii` or `binary`. For additional local interfaces to storage that use the standard FTP daemons underneath, see the NFT (page 12) and HTAR (page 47) sections below. Because the "secure FTP" (SFTP) client talks only to a different (SSHD2) daemon, SFTP is *not* an interface to HPSS at LLNL.

Refer to the HPSS System Administration Guide for information on configuring PFTP.

1.1.2. Parallel FTP (PFTP)

[NOTE: on all LC production machines, the default FTP client is really the parallel interface PFTP, and all files over 1 Mbyte move to or from storage in parallel by default.]

The PFTP client supports normal FTP commands plus extensions (including an LLNL-only PARALLEL toggle command). It is built to optimize FTP performance for storing and retrieving files from HPSS by allowing the data to be transferred in parallel to the client. The interface provided to the user has syntax similar to FTP but with some extensions to allow the user to transfer data to and from HPSS across parallel communication interfaces. PFTP supports transfers either via TCP/IP or IPI-3 over HIPPI. In either case, the FTP client communicates directly with HPSS Movers to transfer data.

The following constraints are imposed by PFTP.

- Pipes are not supported.
- Passive connections are not supported.

- ASCII transfers are not supported over the parallel interface because ASCII transfers insert characters. This makes it impossible to send the data in parallel. Since extra characters are inserted in the stream, there is no way to resolve data placement. Warning: Some FTP implementations default to ascii. If this is the case, it will be necessary to specify binary by entering the bin command.
- PFTP client access is supported only from nodes which support the HPSS PFTP client software.

Refer to the HPSS System Administration Guide for information on configuring PFTP.

1.1.3. Network File System Version 2 (NFS V2)

[NOTE: NFS is NOT currently supported as a storage interface on LC machines.]

The purpose of the NFS V2 server interface is to provide transparent access to HPSS name space object and bitfile data for client systems. Following a mount on the HPSS file system name, the user may access HPSS files using standard function calls and command interfaces.

The code written to implement the NFS V2 Server interface is written to the Network File System Specification, RFC-1094, DDN Network Information Center, SRI International, Menlo Park, Ca. The NFS V2 Server interface and data structures are defined by RFC-1094. Since there are no extensions or modifications to the NFS user interface, no additional interface information is provided in the remainder of this document.

The following constraints are imposed by the HPSS NFS V2 server.

- 2 gigabyte maximum file size. This limitation is imposed by NFS, not HPSS.
- All files created using NFS are stored in a single Class of Service. The Class of Service used by NFS is defined by the HPSS administrator.
- NFS transfers are slower than the other HPSS interfaces, and are therefore not recommended for large file accesses.

Refer to the HPSS Administration Guide for information on configuring NFS V2.

1.1.4. IBM SP Parallel I/O File System (PIOFS) Import/Export

[NOTE: PIOFS Import/Export is NOT currently supported as a storage interface on LC machines.]

PIOFS Import/Export provides data exchange between the SP Parallel I/O File System and other file systems, such as HPSS. In support of Import/Export, the piofsie command is provided. The entire file or pieces of a file defined by lists of source offset, destination, and length may be imported or exported. Several import/export options are supported. These options specify information to create new PIOFS or HPSS files, and hints for file striping.

Refer to Chapter 5 of the Installing, Managing, and Using the IBM AIX Parallel I/O File System document for the list of Parallel I/O File System restrictions, limitations, and deviations.

Refer to the HPSS Administration Guide for information on configuring PIOFS Import/Export. Refer to the HPSS Programmer's Reference, Volume 1 for the PIOFS Import/Export programming interfaces.

1.1.5. NFT (LC Local Interface)

The NFT (Network File Transfer) utility was developed at LC and runs on all open and secure LC production machines (but NOT necessarily on other local machines). NFT relies on standard FTP daemons, and so it inherits many of their file-transfer properties, but its user interface offers several features quite different from FTP:

- A special NFT server preauthenticates all NFT transfers, so all NFT executions are passwordless.
- NFT elaborately tracks and numbers all transfers. It automatically persists if system problems delay storing any file, and it keeps detailed records of your file-storage successes and problems.
- Input from and output to files is easy, and NFT's command syntax (unlike FTP's) lends itself to practical use in scripts and batch jobs.
- Some NFT commands especially facilitate transfers to and from storage (so some users regard NFT as primarily a file-storage rather than a general file-transfer tool).

For a concise, task-oriented summary of how to use NFT commands and features, with annotated typical examples, consult the "Using NFT" section of the EZOUTPUT (URL: <http://www.llnl.gov/LCdocs/ezoutput>) Basic Guide. For a complete analysis of NFT syntax and special features, along with a thorough alphabetical command dictionary, consult the NFT Reference Manual (URL: <http://www.llnl.gov/LCdocs/nft>).

Note also that other local tools use FTP daemons for file transfer to storage but offer nonFTP (in fact, graphical) interfaces. One (for workstations) is XDIR; another (for Macintosh computers) is Fetch.

1.1.6. User Utilities

[NOTE: at LC, these utilities are currently available **ONLY** to storage system administrators, not to general users. See EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for instructions on three other storage tools that *are* available for general users, including LSTORAGE, CHMODSTG, and CHGRPSTG.]

The purpose of the HPSS user utilities is to provide the end user with information such as Access Control List (ACL) definitions and Class of Service definitions. In addition, the ability for a user to change his ACL definitions is provided.

The user utilities consist of these commands:

| | |
|--------|--|
| chacl | change an ACL |
| lsacl | list an ACL |
| lshpss | list information for HPSS (Class of Service list, hierarchy list, storage class list, physical volumes, devices and drives, servers, Movers, and metadata) |

1.2. Storage Concepts

This section defines key HPSS storage concepts which have a significant impact on the usability of HPSS. Configuration of the HPSS storage objects and policies is the responsibility of your HPSS administrator.

1.2.1. Class of Service

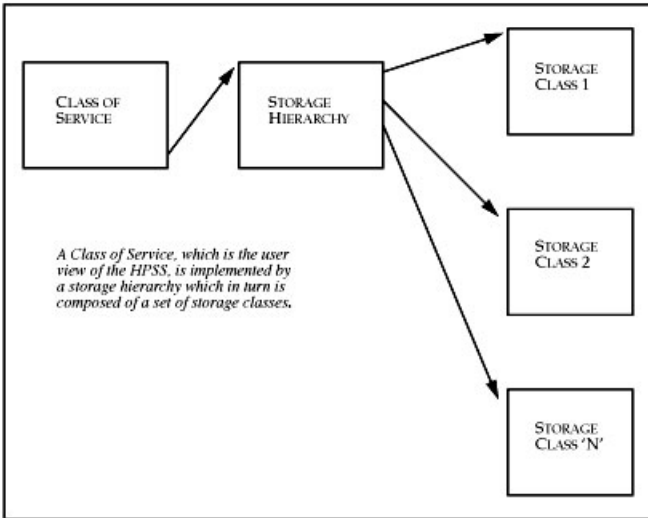
Class of Service (COS) is an abstraction of storage system characteristics that allows HPSS users to select a particular type of service based on performance, space, and functionality requirements. Each COS describes a desired service in terms of characteristics such as minimum and maximum file size, transfer rate, access frequency, latency, and valid read or write operations. A file resides in a particular COS and the class is selected when the file is created. Underlying a COS is a storage hierarchy that describes how data for files in that class are to be stored in HPSS.

For the FTP and PFTP interfaces, the COS ID may be explicitly specified by using the site setcos (page 23) command. If not specified, a default COS is used. [See the Setcos (page 23) section below for (1) a current list of COSs supported at LC and their intended roles, and (2) comments on how LC's NETMON tool monitors and reports FTP traffic to storage.llnl.gov using these same COS categories.]

For NFS, all files are created in the same COS. This COS is defined by your system administrator. For PIOFS Import/Export requests, COS ID may be specified through the -hints option. Contact your HPSS administrator to determine the COSs which have been defined. The `lshpss -cos` command may also be used to list the defined COSs. Refer to Chapter 5 (page 61) for information on the `lshpss` command. [At LC, `lshpss` is NOT available to general users, only system administrators.]

Also, PFTP provides a feature to automatically store the local file size in the minimum and maximum file size fields of the COS. This feature is also provided for FTP clients which support the ALLO command. This allows the COS selection to be made according to file size. The HPSS administrator should ensure that COS definitions contain proper minimum and maximum file sizes in order for PFTP (FTP clients which support ALLO) to optimize storage utilization when transferring files to HPSS. If the COS ID is explicitly set by using the `site setcos` command, that COS will be used regardless of file size. [NOTE: At LLNL, the default Class of Service is in fact dependent on file size except for files written with HTAR.]

A COS is implemented by a Storage Hierarchy of one to many Storage Classes. Storage Hierarchies and Storage Classes are not directly visible to the user, but are described below since they map to COS. The relationship between storage class, storage hierarchy, and COS is shown in this figure:

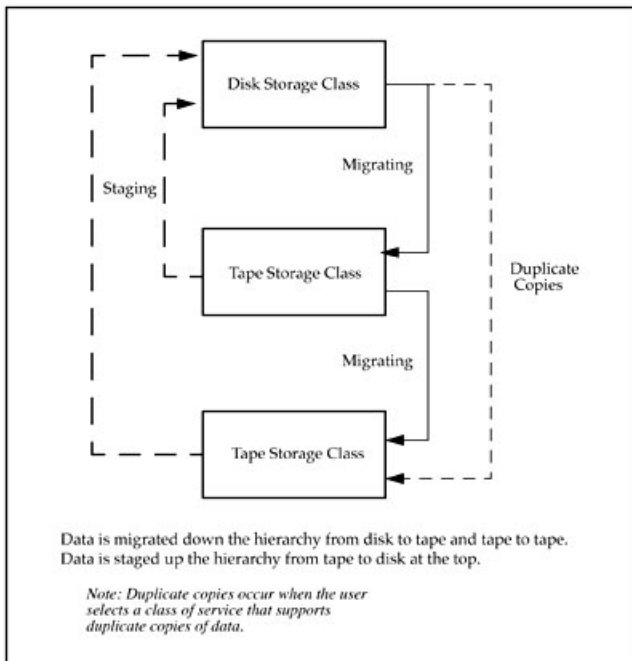


1.2.2. Storage Class

An HPSS Storage Class is used to group storage media together to provide storage with specific characteristics for HPSS data. The attributes associated with a storage class are both physical and logical. Physical media in HPSS are called physical volumes. Physical characteristics associated with physical volumes are the media type, block size, the estimated amount of space on volumes in this class, and how often to write tape marks on the volume (for tape only). Physical media are organized into logical virtual volumes. This allows striping of physical volumes. Some of the logical attributes associated with the storage class are virtual volume block size, stripe width, data transfer rate, latency associated with devices supporting the physical media in this class, and storage segment size (disk only). In addition, the storage class has attributes that associate it with a particular migration policy and purge policy to help in managing the total space in the storage class.

1.2.3. Storage Hierarchy

An HPSS storage hierarchy consists of multiple levels of storage with each level representing a different storage media (i.e., a storage class). Files are moved up and down the storage hierarchy via stage and migrate operations, respectively, based upon storage policy, usage patterns, storage availability, and user request, as shown here:



For example, a storage hierarchy might consist of a fast disk, followed by a fast data transfer and medium storage capacity robot tape system, which in turn is followed by a large data storage capacity, but relatively slow data transfer tape robot system. Files are placed on a particular level in the hierarchy depending on the migration policy and staging operations. Multiple copies of a file may also be specified in the migration policy. If data is duplicated for a file at multiple levels in the hierarchy, the more recent data is at the higher level (lowest level number) in the hierarchy. Each hierarchy level is associated with a single storage class.

See the [SETCOS](#) (page 23) section below for the intended role of each storage class currently offered at LC.

1.3. Interface Usage Considerations

NOTE: at LC, currently only storage interfaces based on the FTP daemons are supported (such as FTP (page 21), NFT (page 12), PFTP, HTAR (page 47), and XDIR). The NFS and PIOFS interfaces are NOT available.

Guidance on when to use a particular HPSS interface is provided below. In general, PFTP provides the best data transfer performance. NFS is the slowest interface, and should not be the interface of choice for large HPSS data transfers.

Conditions in which the user might elect to use FTP are (note that on all LC production machines, FTP defaults to PFTP automatically):

- Utilizes standard FTP interface.
- Users and applications familiar with FTP can access HPSS with the standard command set.
- FTP supports file sizes up to 512 Gbyte for storage (to bundle many small files into a large stored archive, use HTAR (page 47)).
- Supports any FTP client platforms.
- FTP commands may be issued from any vendor nodes with an FTP interface. No specialized code is required.
- FTP (and PFTP) traffic to or from storage from the perspective of several benchmark network nodes (on OCF and SCF) is monitored by LC's NETMON web site and reported using the same "class of service" categories for file size that HPSS uses itself. See the NETMON Reference Manual (URL: <http://www.llnl.gov/LCdocs/netmon>) for details.

Conditions in which a user might elect to use PFTP are (note that on all LC production machines, FTP defaults to PFTP automatically):

- Provides faster file transfers.
- PFTP is a better performer than FTP since it provides the capability to stripe data across multiple client data ports. In addition, IPI-3 transfers to the client are also supported.
- PFTP supports file sizes up to 512 Gbyte for storage (to bundle many small files into a large stored archive, use HTAR (page 47)).
- Supports partial file transfer.
- PFTP provides options on the pget and pput commands to perform partial file transfers. This would be beneficial to users who want to extract pieces of large files.
- On LC production machines, a locally added PARALLEL command lets you toggle parallel transfers on and off while you use regular FTP commands to actually move files.

Conditions in which a user might elect to use NFS are:

- Provides standard system access.
- Files may be accessed and managed through standard system mechanisms without calling a special library or program to translate commands.

- Eliminates multiple file instances.
- The need to maintain multiple instances of a file can be eliminated since files remain on the NFS server.
- Accesses limited to smaller files.
- NFS V2 is limited to 2 gigabyte file sizes, and data transfer performance is slower than the other interfaces.
- Supports any NFS client platforms.
- NFS access is supported from any vendor nodes with an NFS V2 interface. No specialized code is required.

Conditions in which a user might elect to use PIOFS Import/Export are:

- Copy data between the Parallel I/O File System and HPSS.
- PIOFS Import/Export would only be used if data is to be imported into the IBM SP Parallel I/O File System from HPSS, or if data is to be exported from the IBM SP Parallel I/O File System into HPSS.

1.4. User IDs

After the HPSS system is configured, the necessary accounts must be created for HPSS users. Contact your HPSS administrator to add an account.

For FTP or PFTP access, an FTP account must be created. The administrator can use the

hpssuser -add *user* -ftp
command to add a new FTP user.

For NFS access, a DCE user account must be created. The administrator can use the

hpssuser -add *user* -dce
command to add a new DCE account.

Users initiating HPSS imports or exports from or to PIOFS must either:

(1) have a DCE user ID defined which matches their UNIX user ID. The administrator can use the

hpssuser -add *user* -dce
command to add a new DCE user.

OR

(2) have their UNIX password file entry imported into the security registry. The administrator can use the `passwd_import` command to import a password file into the security registry.

WARNING: Users calling the utilities described in this document must be logged into DCE. As noted above, the administrator can use the

hpssuser -add *user* -dce
command to add a new DCE account.

1.5. DCE User Accounts

As mentioned in the previous section, the user utilities and PIOFS Import/Export may require the user be logged into DCE.

The following command is used to issue a DCE login:

```
dce_login [principal_name] [password]
```

When this command is entered, the principal's identity is validated, and the network credentials are obtained. If principal name or password are not supplied, dce_login will prompt for them.

When the principal's DCE login context is no longer required, the following command may be used to destroy the login context and associated credentials:

```
kdestroy
```

Other DCE commands which might be of interest to the user are:

| | |
|-------|---|
| klist | list the primary principal and tickets held in the DCE credentials cache. |
| kinit | refresh a DCE credentials cache. |

2. File Transfer Protocol (FTP)

This chapter specifies the HPSS FTP interface. For information on configuring the HPSS FTP daemon, refer to the HPSS Administration Guide. FTP is supported from any FTP client platform (but see the limitation in the next paragraph and the warning about HPSS server differences farther below).

WARNING: LC now uses its hardware/software security "firewall" to block direct FTP connections from machines outside the llnl.gov domain to LC machines within llnl.gov (including storage). This firewall blocking of incoming FTP connections means that if you have files on any machine outside llnl.gov and you want to store them in HPSS, you must either:

- (1) first log on to a machine within llnl.gov, run FTP there to transfer (GET) your files to that within-llnl.gov machine, and then transfer (PUT) them again from there to storage, or
- (2) Before you run FTP on your outside-the-firewall machine, get, install, configure, and execute a Virtual Private Network (VPN) client on that machine. Contact the LC Hotline to see if you are authorized to run a VPN client for access to LLNL. A VPN client borrows an llnl.gov IP address for your machine while it runs, and LC has confirmed that if you run VPN and FTP together under Windows98, you can directly transfer files to storage.llnl.gov from outside the firewall (no staging to an LC production machine is needed). But you may encounter vendor-compatibility problems with other versions of Windows or with other operating systems. See LC's Firewall and SSH Guide (URL: <http://www.llnl.gov/LCdocs/firewall>) for full instructions on the fairly complex process of getting and using VPN to enable FTP.

For information on additional user interfaces that also rely on the FTP daemons underneath, see the NFT (page 12) and HTAR (page 47) sections below. Because the "secure FTP" (SFTP) client talks to a different (SSHD2) daemon, SFTP is *not* an interface to HPSS on LC machines.

HPSS supports the FTP command set for transferring files to and from HPSS. To use FTP, the user enters the following:

```
ftp node_name [port_number]
```

where,

node_name is the node name of the node where the HPSS FTP Daemon process resides.

port_number is the port number for HPSS, as set up in /etc/services.

At this point, any standard FTP command may be entered. Note: If the message "Load thread state failed" is received, contact your HPSS administrator. This message generally implies that either HPSS is not correctly configured, or some HPSS components may not be executing.

Different FTP servers process incoming FTP "M" commands (such as MDELETE) differently, with different file-transfer results. The FTP server(s) on HPSS at LC are unusual in interpreting MDELETE filters (such as TEST*) recursively; they remove all (matching) files not only in the current working directory but also in the directory children of that directory as well. To minimize the impact of this aggressive interpretation of MDELETE on your stored files, see the "FTP Pitfalls (with Storage)" section of the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) guide.

On all LC production machines, typing FTP automatically executes the locally enhanced PFTP (parallel FTP) client, and all files over 1 Mbyte automatically move to or from storage by parallel file transfer (unless

you explicitly request otherwise). See the FTP Reference Manual (URL: <http://www.llnl.gov/LCdocs/ftp>) for how to force execution of the serial FTP client and for general advice on using parallel FTP service effectively in the LLNL computing environment.

2.1. Site Commands

HPSS also supports the site (supplementary) commands for FTP listed below.

- setcos
- chgid
- chgrp
- chmod
- chown
- chuid
- stage
- wait

These commands differ substantially from the supplementary commands formerly supported by the UniTree storage system (page 69), and even when they have the same name (e.g., **wait**) they often have different roles. Some FTP clients let you type (some of) these commands without a prefix (e.g., **chmod**), some clients expect the **site** prefix (as shown in this manual), and some clients require the longer **quote site** prefix (e.g., **quote site stage**). Only testing will reveal which format your FTP client requires on a specific machine, although the **quote site** long form is always an acceptable fall back.

Details for each supplementary command appear in the following subsections.

2.1.1. SETCOS (Specifying a File's Class of Service)

SETCOS is used to specify a class of service and has the following format:

```
site setcos cos_id
```

where,

cos_id is the Class of Service identifier (used when creating a new HPSS file during a put operation). WARNING: HPSS does not verify the *cos_id* that you supply, and it now accepts without error any integer as a *cos_id*, even if no such COS exists.

Class of Service is used as a means for specifying how the storage system handles the incoming files. For example, COS may determine the number of copies kept or the amount of parallelism or stripe width for a file. If a Class of Service is not specified, a default is used (and using the default COS at LC is recommended for maximum efficiency).

At LC, HPSS currently supports the following Classes of Service on both the OCF and SCF storage systems:

| COS ID | Default for the file size(*) | Copies kept | Slang term |
|--------|------------------------------|-------------|---------------|
| 110 | 0 .GE. file .LT. 4 Mbyte | 2(*) | small (sFTP) |
| 120 | 4 .GE. file .LT. 32 Mbyte | 2(*) | medium (mFTP) |
| 130 | 32 .GE. file .LT. 256 Mbyte | 1 | large (lFTP) |
| 140 | 256 .GE. file | 1 | jumbo (jFTP) |
| 150 | [by request only] | 2 | |

(*)For files stored using FTP or NFT. All files stored using HTAR, regardless of size, have COS 140 (and hence single-copy storage) by default. COS 150 allows users to overtly request dual-copy storage even for large files or for files of any size stored using HTAR (for mission critical files only, please).

LC's goal is for the FTP storage server to automatically detect incoming file size (and sending client) and assign a Class of Service appropriately, with no intervention by the user. COS details at LC may change with time to optimize storage system performance.

In the example below, the following command might be entered to put a large file into HPSS with a Class of Service ID 150, to get the protection of dual-copy storage for a large but mission critical file:

```
site setcos 150
```

NETMON, LC's network-monitoring web site, tracks FTP traffic to (or from) HPSS from the perspective of several benchmark OCF and SCF network nodes. NETMON tracks each HPSS Class of Service separately. In fact, NETMON's tables and graphs *always* report FTP traffic by using HPSS Class of Service categories for file size even when storage.llnl.gov is not one of the nodes generating that FTP traffic (see the NETMON COS "slang" terminology in the right-most column of the chart above). The NETMON Reference Manual (URL: <http://www.llnl.gov/LCdocs/netmon>) tells how to interpret and customize NETMON's FTP-monitoring reports.

2.1.2. CHGID (Changing a File's Group by ID)

CHGID is used to change the group ID of a file and has the following format:

```
site chgid gid file
```

where,

gid is the new group ID of the file.

file is the name of the file.

The user must belong to the specified group and be the owner of the file, or be the root user.

Example: The following may be entered to change the group ID of myfile to group ID 210.

```
site chgid 210 myfile
```

2.1.3. CHGRP (Changing a File's Group by Name)

CHGRP is used to change the group name of a file and has the following format:

```
site chgrp grp file
```

where,

grp is the new group name of the file.

file is the name of the file.

The user must belong to the specified group and be the owner of the file, or be the root user. (For an alternative way to change a stored file's group, see the instructions for the CHGRPSTG tool in the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide.)

Example: The following may be entered to change the group of myfile to group mygroup.

```
site chgrp mygroup myfile
```

2.1.4. CHMOD (Changing a File's Permissions)

CHMOD is used to change the group mode (permissions) of a file and has the following format:

```
site chmod mode file
```

where,

mode is the new octal mode number of the file. Symbolic CHMOD parameters (such as u or g) are not supported by the FTP CHMOD command. For a good short review of the octal permission notation, see [EZFILES](http://www.llnl.gov/LCdocs/ezfiles) (URL: <http://www.llnl.gov/LCdocs/ezfiles>). (For an alternative way to change a stored file's permissions (octal or symbolic), see the instructions for the CHMODSTG tool in [EZSTORAGE](http://www.llnl.gov/LCdocs/ezstorage) (URL: <http://www.llnl.gov/LCdocs/ezstorage>).)

file is the name of the file.

Mode is constructed from the OR of the following modes:

```
0400 read by owner
0200 write by owner
0100 execute (search in a directory) by owner
0040 read by group
0020 write by group
0010 execute (search in a directory) by group
0004 read by others
0002 write by others
0001 execute (search in a directory) by others
```

Note: The following mode values are not supported:

```
4000 set user ID on execution
2000 set group ID on execution
1000 sticky bit
```

Only the owner of the file or root user can change its mode.

Example: The following may be entered to change the mode of myfile to read, write by owner and group.

```
site chmod 0660 myfile
```

2.1.5. CHOWN (Changing a File's Owner by Name)

CHOWN is used to change the owner of a file and has the following format:

```
site chown owner file
```

where,

owner is the new owner of the file.

file is the name of the file.

Only the root user can change the owner of a file.

Example: The following may be entered to change the owner of /home/smith/myfile to jones.

```
site chown jones /home/smith/myfile
```

2.1.6. CHUID (Changing a File's Owner by ID)

CHUID is used to change the uid (user ID) of a file and has the following format:

```
site chuid uid file
```

where,

uid is the new uid of the owner of the file.

file is the name of the file.

Only the root user can change the uid of a file.

Example: The following may be entered to change the uid of /home/smith/myfile to 201.

```
site chown 201 /home/smith/myfile
```

2.1.7. STAGE (Staging a File)

STAGE is used to initiate a stage of a migrated file (e.g. from tape to disk). The user can initiate the stage and then return at a later time to initiate the file transfer using the FTP get or PFTP pget commands:

```
site stage file
```

where,

file is the name of the file.

Example: The following may be entered to stage file /home/smith/myfile.

```
site stage /home/smith/myfile
```

2.1.8. WAIT (Setting the Desired Wait Options for Migrated Files)

WAIT is used to notify the HPSS PFTP Daemon :

site wait *option*

where,

option is one of the following values:

- | | |
|-----------------------|---|
| -1 or inf[inite] | wait forever for the file to be staged. Do not return from the get or pget command to complete until the file has been transferred or a transfer error has occurred. |
| 0 | do not wait for the file to be staged. If the file has been migrated, return the appropriate message and initiate the stage. The user will return later to reissue the get or pget command. |
| <i>n</i> (an integer) | wait the specified period (in seconds) for the file requested by a get or pget command to complete. Either transfer the file if the file is staged within the specified period or return a reply to notify the user to try again later. |

Example: The following may be entered to wait for files to be staged.

```
site wait -1
```

The following table describes the behaviour the customer should expect from FTP when issuing the stage/wait commands. Note: ONLY Classes of service utilizing the "Stage on Background" will exhibit predictable results.

Table 2. Stage/Wait Behaviour

| Wait Time | File Condition | Command | Behavior/Message |
|-----------------|----------------|----------------|---|
| No Wait | Archived | site stage xyz | "File xyz is being retrieved from archive." |
| No Wait | Not Archived | site stage xyz | "File xyz is currently ready for other processing." |
| Wait <i>nnn</i> | Archived | site stage xyz | Wait for period then receive message: "File xyz is currently ready for other processing." or "File xyz is currently ready for other processing." if the file is staged in the time frame allowed. |
| Wait <i>nnn</i> | Not Archived | site stage xyz | "File xyz is currently ready for other processing." |
| No Wait | Archived | get xyz | "File xyz is being retrieved from archive." |
| No Wait | Not Archived | get xyz | Transfers Data as expected. |
| Wait <i>nnn</i> | Archived | get xyz | Wait for period then receive message: "File xyz is being retrieved from archive." or transfers data as expected if file is staged in the time allowed. |
| Wait <i>nnn</i> | Not Archived | get xyz | Transfers file as expected. |

2.2. List Directory Extensions

FTP supports the `ls` command to list the contents of a directory. Standard options supported are: `ls`, `ls -l`, `ls -a`, and `ls -F`. In addition to the standard `ls` options generally provided, HPSS also provides a `-lh` option. If `-lh` is specified, then a long directory listing is generated. However, in place of the owner field (field 3) and group field (field 4) listed for the `-l` option, the Class of Service identifier and Account Code are listed.

Example: `ls -lh`

| | | | | | |
|-------------------------|------------------|------------------|------------------------|--------------------------|----------------------|
| <code>-rw-rw----</code> | <code>1 1</code> | <code>198</code> | <code>157286400</code> | <code>May 13 1996</code> | <code>TEST</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod1</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod10</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod11</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod12</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod13</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod14</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod15</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod151</code> |
| <code>-rw-r--r--</code> | <code>1 1</code> | <code>160</code> | <code>32768</code> | <code>May 16 1996</code> | <code>prod152</code> |

For an alternative way to list stored files and their attributes, including recursive listings of all or part of your storage hierarchy easily redirected to a file, see the instructions for the `LSTORAGE` tool in the [EZSTORAGE](http://www.llnl.gov/LCdocs/ezstorage) (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide.

3. Parallel File Transfer Protocol (PFTP)

Whenever you run FTP on LC production machines (OCF and SCF), but not necessarily on special-purpose machines, you are automatically executing a customized PFTP client. For instructions on avoiding PFTP, for advice on how PFTP interacts with the jumbo-frame (high-capacity) links on some LC machines, and for information on the PARALLEL command added to the local implementation of PFTP, see the "Parallel FTP Service" section of the FTP Reference Manual (URL: <http://www.llnl.gov/LCdocs/ftp>). The text below describes the *standard* PFTP interface, which is available on LC production machines but largely unnecessary if you use the local PARALLEL command instead (see 3.1.10 below (page 46)). However, if you try to store files at other (tri-lab) sites where LC's local customizations are missing, then you can still use the native PFTP commands listed here to perform parallel file transfers.

This chapter specifies the HPSS PFTP interface. For information on configuring the HPSS PFTP daemon, reference the HPSS Administration Guide. In order to use PFTP, the PFTP client code must be supported from the client platform.

PFTP supports the FTP command set plus some additional commands (refer to the next subsection). To use PFTP, the user enters one of the following commands:

```
pftp_client node_name [port_name]
```

```
pftp_client_ipi3 node_name [port_name]
```

where,

node_name is the node name of the node where the HPSS PFTP Daemon process resides.

port_number is the port number for HPSS, as set up in /etc/services.

The `pftp_client_ipi3` command would be used when file transfers use the IPI-3 protocol. The local administrator may opt to define a `pftp` program link that points to either `pftp_client` or `pftp_client_ipi3`, as appropriate. [At LLNL, IPI-3 is not used, and the full pathname of the PFTP client is /usr/local/bin/pftp.]

Two additional forms of the `pftp_client` are available (upon request and pursuant to legal restrictions): `krb5_gss_pftp_client` and `dce_gss_pftp_client`. Since the Generic Security Service (GSS) versions of the Parallel FTP Client only relate to the authentication process, the HIPPI (IPI3) forms of these applications should behave the same as the non-GSS versions. The GSS clients are used to provide credential authentication facilities (password-less authentication) between the client and the HPSS GSS Parallel FTP Daemon using either Kerberos or DCE credentials for authentication.

The GSS versions of the Parallel FTP Client require either Kerberos and/or DCE Client software and the Data Encryption Standard (DES) libraries. The Kerberos version is based on MIT Kerberos Beta Version 5.6. Updates are anticipated in the future; however, neither IBM nor the HPSS Development team are obligated to pursue this path.

As a courtesy to HPSS customers, the Parallel FTP Client code is available for compilation at customer sites upon request. The Parallel FTP Client code will be provided as a .tar.Z file (tarred and compressed) containing all components required to build the applications. Hardware/Software dependencies are the individual HPSS customers responsibility. This explicitly denies any support requirement on IBM or the

HPSS Development/Support personnel for any modifications made by the customer. No DCE software is required to build the HPSS Parallel FTP Client.

The HPSS PFTP Client has been successfully compiled on: Cray UNICOS, Hewlett-Packard HPUX, Silicon Graphics IRIX (32 Bit), Sun Solaris, Intel Paragon OSF, Intel Teraflop OSF, and IBM AIX 4.x. Ports to other hardware/software components are the responsibility of the remote site. These sites will be asked to share their ports with the HPSS development team (and other HPSS facilities); however, neither IBM nor the HPSS Development Team accepts any obligation to incorporate any hardware/software ports into the distribution source. No site specific features (local mods) added to the Parallel FTP client by customer sites will be incorporated into the PFTP client without the modification of the HPSS license.

The GSS versions of the Parallel FTP Client require MIT Kerberos and/or the DCE Toolkit. The Kerberos V5B6 code is available upon request (pursuant to restrictions). Neither IBM nor the HPSS Development team declare the Kerberos code suitable for any specific purpose nor are they obligated to repair or support customers using this code. The GSS PFTP Client code has been compiled on IBM AIX 4.x and Sun Microsystems Solaris 5.4.

The GSS HPSS Parallel FTP Daemon is available for IBM AIX 4.x only.

Note: If the message "Load thread state failed" is received, contact your HPSS administrator. This message generally implies that either HPSS is not correctly configured, or some HPSS components may not be executing.

3.1. Additional Commands

All FTP extensions described in Chapter 2 are supported by PFTP. In addition, the following commands are supported by PFTP.

- pappend
- pput, mpput
- pget, mpget
- psocket
- pipi3
- setpwidth
- setpblocksize
- parallel [LLNL only] (page 46)
- HTAR [LLNL-only front end for PFTP] (page 47)

3.1.1. PAPPEND (Parallel append)

SYNOPSIS.

```
pappend local_file [remote_file]
```

DESCRIPTION.

The pappend command transfers a file from the local machine to HPSS. The transfer starts at the end of the remote file and continues until the entire file is moved or until an error occurs.

PARAMETERS.

| | |
|--------------------|---|
| <i>local_file</i> | Identification of the file to transfer on the local machine. |
| <i>remote_file</i> | Optional file name to the remote file. If not supplied then the remote (HPSS) file name defaults to be the same as the local file name. |

RETURN STRINGS.

Output shows the amount of data transferred and any error conditions.

ERROR CONDITIONS.

Connection Failures: data transfer connection malfunction.

Network Failures: data transfer malfunction.

Allocation Failures: no space on remote machine for file.

Error codes may also be returned from HPSS. The most common error codes are:

-5....an I/O error occurred.

-28...no space remaining in the associated storage class.

SEE ALSO.

RFC-0959.

NOTES.

In LC's current implementation of HPSS, a file's "Class of Service" when stored depends on its size. Files below 4 Mbyte are in one Class of Service and those above 4 Mbyte are in another. If you store a file that falls just below the 4-Mbyte threshold and then use PAPPEND to write to the end of it, causing it to expand into another Class of Service, HPSS (wrongly) rejects the file transfer and PFTP returns the error message "529 Bad Data Transfer (moved=0)." LC is seeking solutions to this known HPSS bug.

At LC, using PAPPEND is just the same as using the standard FTP APPEND command if "parallel mode" is enabled. On LC production machines it is automatically enabled for transfers over 1 Mbyte to or from storage (and you can toggle it by using LC's local PARALLEL (page 46) command).

EXAMPLES.

1. Append local file testfile to the same file name in the user's HPSS home directory.

```
pappend testfile
```

2. Append local file testfile to HPSS file prod1 in the current working directory.

```
pappend testfile prod1
```

3.1.2. PPUT (Parallel file store)

SYNOPSIS.

```
pput [-l local_offset] [-r remote_offset]  
  
[-s size] local_file [remote_file]
```

DESCRIPTION.

The pput command transfers a file from the local machine to HPSS. If offsets and size of transfer are not specified, the transfer starts at the beginning of the local file and continues until the entire file is moved or until an error occurs. However, flexibility is provided to perform partial file transfers by specifying local file offset, remote file offset, and size of transfer.

The normal pput command functions just like the standard FTP put command and transfers an entire file. [LC users see the NOTES below.]

PARAMETERS.

-l local_offset Optional byte offset into the local file where the transfer is to begin.

-r remote_offset
 Optional byte offset into the remote file where the data is to be placed.

-s size Optional number of bytes of data to transfer.

local_file Identification of the file to transfer on the local machine.

remote_file Optional file name to the remote file. If not supplied then the remote (HPSS) file name defaults to be the same as the local file name.

RETURN STRINGS.

Output shows the amount of data transferred and any error conditions.

ERROR CONDITIONS.

Connection Failures: data transfer connection malfunction.

Network Failures: data transfer malfunction.

Allocation Failures: no space on remote machine for file.

Error codes may also be returned from HPSS. The most common error codes are:

-5....an I/O error occurred.

-28...no space remaining in the associated storage class.

SEE ALSO.

RFC-0959.

NOTES.

At LC, using PPUT is just the same as using the standard FTP PUT command if "parallel mode" is enabled. On LC production machines it is automatically enabled for transfers over 1 Mbyte to or from storage (and you can toggle it by using LC's local PARALLEL (page 46) command).

EXAMPLES.

1. Transfer local file testfile to the same file name in the user's HPSS home directory.

```
pput testfile
```

2. Transfer local file testfile to HPSS file prod1 in the current working directory.

```
pput testfile prod1
```

3. Transfer 1 MB from offset 1 MB of local file testfile to offset 0 of HPSS file /home/bob/prod1.

```
pput -l 1048576 -r 0 -s 1048576 testfile /home/bob/prod1
```

3.1.3. MPPUT (Parallel file store)

SYNOPSIS.

mpput *local_files*

DESCRIPTION.

The mpput command expands the files specified in the *local_files* parameter at the local host and copies the indicated files to HPSS. The mpput command functions just like the standard FTP mput command. [LC users see the NOTES below.]

PARAMETERS.

local_files Identification of the files to transfer on the local machine.

RETURN STRINGS.

Output shows the amount of data transferred and any error conditions.

ERROR CONDITIONS.

Connection Failures: data transfer connection malfunction.

Network Failures: data transfer malfunction.

Allocation Failures: no space on remote machine for file.

Error codes may also be returned from HPSS. The most common error codes are:

-5....an I/O error occurred.

-28...no space remaining in the associated storage class.

SEE ALSO.

RFC-0959.

NOTES.

At LC, using MPPUT is just the same as using the standard FTP MPUT command if "parallel mode" is enabled. On LC production machines it is automatically enabled for transfers over 1 Mbyte to or from storage (and you can toggle it by using LC's local PARALLEL (page 46) command).

EXAMPLES.

1. Transfer all local files in the current directory to the user's HPSS home directory.

mpput *

2. Transfer all local files that begin with TEST in directory /usr/bob to the user's HPSS home directory.

mpput /usr/bob/test*

3.1.4. PGET (Parallel file retrieval)

SYNOPSIS.

```
pget [-r remote_offset] [-l local_offset]  
  
[-s size] remote_file [local_file]
```

DESCRIPTION.

The pget command transfers a file to the local machine from HPSS. If offsets and size of transfer are not specified, the transfer starts at the beginning of the remote file and continues until the entire file is moved or until an error occurs. However, flexibility is provided to perform partial file transfers by specifying local file offset, remote file offset, and size of transfer.

The normal pget command functions just like the standard FTP get command and transfers an entire file. [LC users see the NOTES below.]

PARAMETERS.

| | |
|-------------------------|---|
| <i>-r remote_offset</i> | Optional byte offset where transfer is to begin in the remote file. |
| <i>-l local_offset</i> | Optional byte offset into the local file where the data is to be placed. |
| <i>-s size</i> | Optional number of bytes of data to transfer. |
| <i>remote_file</i> | Identification of the file to transfer from the remote (HPSS) machine. |
| <i>local_file</i> | Optional file name to the local file. If not supplied then the local file name defaults to be the same as the remote file name. |

RETURN STRINGS.

Output shows the amount of data transferred and any error conditions.

ERROR CONDITIONS.

Connection Failures: data transfer connection malfunction.

Network Failures: data transfer malfunction.

Allocation Failures: no space on remote machine for file.

Error codes may also be returned from HPSS. The most common error codes are:

-5....an I/O error occurred.

SEE ALSO.

RFC-0959.

NOTES.

At LC, using PGET is just the same as using the standard FTP GET command if "parallel mode" is enabled. On LC production machines it is automatically enabled for transfers over 1 Mbyte to or from storage (and you can toggle it by using LC's local PARALLEL (page 46) command).

EXAMPLES.

1. Transfer HPSS file /home/bob/prod1 to the same file name in the user's local directory.

```
pget /home/bob/prod1
```

2. Transfer HPSS file prod1 to local file testfile1 in the current working directory.

```
pget prod1 testfile1
```

3. Transfer 1 MB from offset 0 of HPSS file /home/bob/prod1 to offset 1048576 of local file testfile.

```
pget -l 1048576 -r 0 -s 1048576 /home/bob/prod1 testfile
```

3.1.5. MPGET (Parallel file retrieval)

SYNOPSIS.

mpget *remote_files*

DESCRIPTION.

The mpget command expands the files specified in the *remote_files* parameter at the remote (HPSS) host and copies the indicated HPSS files to the current directory on the local host. The mpget command functions just like the standard FTP mget command. [LC users see the NOTES below.]

PARAMETERS.

remote_files Identification of the files to transfer from the remote (HPSS) machine.

RETURN STRINGS.

Output shows the amount of data transferred and any error conditions.

ERROR CONDITIONS.

Connection Failures: data transfer connection malfunction.

Network Failures: data transfer malfunction.

Allocation Failures: no space on remote machine for file.

Error codes may also be returned from HPSS. The most common error codes are:

-5....an I/O error occurred.

SEE ALSO.

RFC-0959.

NOTES.

At LC, using MPGET is just the same as using the standard FTP MGET command if "parallel mode" is enabled. On LC production machines it is automatically enabled for transfers over 1 Mbyte to or from storage (and you can toggle it by using LC's local PARALLEL (page 46) command).

EXAMPLES.

1. Transfer all files in the HPSS directory /home/bob to the user's current local directory.

mpget /home/bob/*

2. Transfer all HPSS files that begin with TEST in directory /usr/bob to the user's current local directory.

mpget /usr/bob/test*

3.1.6. P SOCKET (Specify TCP socket based transfers)

SYNOPSIS.

psocket

DESCRIPTION.

The psocket command is used to specify to the FTP client code that any parallel transfers are now to be done using connection based sockets (TCP).

PARAMETERS.

none.

RETURN STRINGS.

"Parallel transfers will not go over sockets."

ERROR CONDITIONS.

none.

SEE ALSO.

RFC-0959.

NOTES.

If neither psocket nor pipi3 is specified, psocket is the default. On LC production machines, psocket is indeed the default and the only available choice, so this command is entirely redundant.

3.1.7. PIPI3 (Specify IPI-3 based transfers)

SYNOPSIS.

pipi3

DESCRIPTION.

The pipi3 command is used to specify to the FTP client code that any parallel transfers are now to be done using the IPI-3 protocol. This protocol allows the PFTP client to transfer data directly to or from a HIPPI-attached device.

PARAMETERS.

none.

RETURN STRINGS.

"Parallel transfers will not go over ipi3."

ERROR CONDITIONS.

If IPI-3 is not supported, then the following will be printed:

"IPI-3 tranfers not allowed"

SEE ALSO.

RFC-0959.

NOTES.

This option is only appropriate for the pftp_client_ipi3 command. If neither psocket nor pipi3 is specified, psocket is the default. On LC production machines, psocket is indeed the default and the only available choice, so pipi3 always returns the message "IPI3 transfers not allowed."

3.1.8. SETPWIDTH (Specify transfer stripe width)

SYNOPSIS.

setpwidth *stripe_width*

DESCRIPTION.

The setpwidth command is used to specify the size of the client-side stripe to the FTP client code.

PARAMETERS.

stripe_width The width of the PFTP client-side stripe. The width can have a value of 1 through 16. The stripe width from the PFTP client perspective is the number of client processes spawned to handle the data transfers. Stripe width from the server perspective is the number of volumes the file is striped across.

On LC production machines the default stripe width for files under 1 Mbyte moving to or from storage is 1. The default stripe width for files 1 Mbyte or larger is 4, chosen to optimize most file transfers.

A general guideline would be to set *stripe_width* to an even divisor of the number of volumes the file is striped across. For example, if the Class of Service for a file were set up for a 4-way stripe, suggested values for *stripe_width* might be 2 or 4.

If the stripe width of the file is unknown, consult your HPSS administrator or follow the following steps to determine the stripe width:

1. Enter the `lshpss -cos` command to list Class of Service information. From the entry with the CID value equal to your Class of Service ID, locate the HID (hierarchy ID) field.
2. Enter the `lshpss -hier` command to list hierarchy information. From the entry matching the HID value above, locate the StorageClassID field.
3. Enter the `lshpss -sc` command to list the storage class information. From the entry matching the StorageClassID value above, locate the W field. This is the stripe width for the storage class of the file.

RETURN STRINGS.

"Parallel stripe width set to [*stripe width*]."

ERROR CONDITIONS.

"Bad width value [*stripe width*]."

SEE ALSO.

RFC-0959.

NOTES.

On LC production machines, the default stripe width of 4 (for files 1 Mbyte or larger) has been chosen to optimize most file transfers to or from storage. Most users will only decrease the effectiveness of their storage transfers by using SETPWIDTH to change to another stripe width.

EXAMPLES.

1. Set the stripe width to 2.

```
setpwidth 2
```

3.1.9. SETPBLOCKSIZE (Specify transfer block size)

SYNOPSIS.

setpblocksize *block_size*

DESCRIPTION.

The setpblocksize command is used to specify the block size to be used for parallel transfers.

PARAMETERS.

block_size The number of bytes to be transferred to each element of the stripe before data is sent to the next element. The current allowable transfer sizes range from 1 through 16 MB. The default block size for transfers to or from storage on LC production machines is 1 Mbyte (1048576 bytes), not the 256 Kbyte used at some other HPSS sites.

A general guideline would be to set *block_size* to the virtual volume block size. Consult your HPSS administrator or follow the following steps to determine the virtual volume block size:

1. Enter the `lshpss -cos` command to list Class of Service information. From the entry with the CID value equal to your Class of Service ID, locate the HID (hierarchy ID) field.
2. Enter the `lshpss -hier` command to list hierarchy information. From the entry matching the HID value above, locate the StorageClassID field.
3. Enter the `lshpss -sc` command to list the storage class information. From the entry matching the StorageClassID value above, locate the VVBlk field. This is the virtual volume block size.

RETURN STRINGS.

"Parallel block size set to [block size]."

ERROR CONDITIONS.

"Bad block size value [block size]."

SEE ALSO.

RFC-0959.

NOTES.

You may be able to significantly increase the parallel transfer rate for large files by increasing your block size with SETPBLOCKSIZE.

EXAMPLES.

1. Set the transfer block size to 8 MB.

```
setpblocksize 8388608
```

3.1.10. PARALLEL (LLNL only)

SYNOPSIS.

parallel

DESCRIPTION.

The PARALLEL command is a local customization of the PFTP client on LC production machines, and is not available in the storage interface at nonLLNL (tri-lab) sites. The PARALLEL command enables parallel file transfers on LC production machines if they do not already occur automatically. Parallel transfers are ON by default whenever you move a file over 1 Mbyte to or from storage or to any AIX or OCF Tru64 node. But parallel transfers are OFF by default between any other pair of production machines, including Linux machines (so typing PARALLEL once in this case enables them).

PARAMETERS.

None. This command has no arguments.

RETURN STRINGS.

"Auto parallel is [on|off]."

ERROR CONDITIONS.

None.

SEE ALSO.

LC's FTP Reference Manual (URL: <http://www.llnl.gov/LCdocs/ftp>), "Parallel FTP Service" section discusses how parallel transfers interact with jumbo-frame (high-capacity) Ethernet links.

NOTES.

LLNL's local PARALLEL command is intended to make the specialized PFTP commands described in the previous subsections unnecessary. On LC production machines, once parallel mode is enabled (by default or by invoking PARALLEL), then you can use standard FTP commands to GET or PUT single or multiple files and automatically enjoy all the benefits of optimized parallel transfers (especially to or from storage) without learning any special PFTP command syntax. PARALLEL also reports the current stripe width and block size. Parallel transfers persist until you end your current FTP session.

EXAMPLES.

1. Enable parallel transfers (IF not on by default).

parallel

3.1.11. HTAR (PFTP Front End, LLNL only)

On LC production machines only (not at other ASCI sites), HTAR is a separate, locally developed utility program that serves as a special-purpose front end to PFTP for storage access. HTAR combines a flexible file bundling tool with fast parallel access to archival storage (HPSS), to let you store and selectively retrieve even very large sets of files very efficiently. (With HTAR's -F option you can adapt this tool for file transfers to LC *nonstorage* hosts too.)

HTAR's enhanced features include:

- Uses a TAR-like syntax and supports TAR-compatible archive files by relying on the POSIX 1003.1 TAR file format.
- Bundles files in memory using multiple concurrent threads and transfers them into an archive file built *directly* in storage by default, to avoid needing extra online disk space.
- Takes advantage of available parallel interfaces to storage to provide fast file transfers (measured at as high as 150 Mbyte/s, which exceeds 30 times the typical rate for transferring small files separately).
- Uses an external index file to easily accommodate thousands of small files in any archive, and to support retrieval of specified files from within a still-stored archive without first retrieving the whole archive from HPSS.
- Imposes no limit on the total size of the archives that it builds (some have reached 200 Gbyte successfully) and accepts input files (archive members) as large as 8 Gbyte.

When the storage system (HPSS) is up and available to users you can execute HTAR with a command line that has the general form

```
htar action archive [options] [filelist]
```

and the specific form

```
htar -c|t|x|X -f archivename [-BdEFhHILmMopSTvVwY] [flist]
```

where exactly one action and the *archivename* are always required, while the control options and (except when using -c) the *filelist* (or *flist*) can be omitted (and the options can share a hyphen flag with the action for convenience). Users familiar with TAR can guess how to run HTAR from this model (although there are some tricky syntax differences). Others should consult the [HTAR Reference Manual](http://www.llnl.gov/LCdocs/htar) (URL: <http://www.llnl.gov/LCdocs/htar>) for usage suggestions, annotated examples, technical tips, full option details, and known problems.

4. IBM SP PIOFS Import/Export

[NOTE: PIOFS Import/Export is NOT currently supported as a storage interface on LC machines.]

This chapter specifies the SP PIOFS Import/Export interfaces. Specifically, the following information is provided:

- Command Line Interface
- Usage of the piofsie command is described in this section.
- Import/Export Parameter File Layout
- The format of an optional file which specifies the import/export options is defined in this section.
- Configuration and Setup
- Required user configuration tasks are defined in this section.

4.1. Command Interface

4.1.1. PIOFSIE (Import or export a file between PIOFS and HPSS)

SYNOPSIS.

piofsie

`-op {IMPORT | EXPORT}`

`-efs efs_name`

`-ename ext_file`

`[-eflags eflag]`

`[-eperms eperm]`

`-pname piofs_file`

`[-pflags pflag]`

`[-pperms pperm]`

`[-precsiz recsize]`

`[-pcells cells]`

`[-vbs vbsn]`

`[-vn vnn]`

`[-hbs hbsn]`

`[-hn hnn]`

`[-subfile subfilen]`

`[-chkptflag {CHECKPOINT | ACTIVE}]`

`[-list src_offset dest_offset requested_len]`

`[-pfile parm_file]`

`[-hints cos_id]`

DESCRIPTION.

The piofsie command may be used to request a PIOFS import or export operation. Note: The name of a temporary parameter options file is sent to the import/export program. The directory name in which the temporary file is created is taken from either the pfile name (if specified in the command) or from the current working directory. The import/export program must have access to this directory. If remote, file system access such as NFS may be required.

PARAMETERS.

`-op {IMPORT | EXPORT}`

The operation to be performed, either IMPORT or EXPORT.

`-efs efs_name` The external file system name. This name is be matched with a name in the PIOFS I/E configuration file to determine the IP address of a node which has an import/export program that executes this command. The file system name may be one to twelve characters. For HPSS, the *efs_name* value must be hpss.

`-ename ext_file`

The full path name of the external file (in the external file system) which is the target of this import or export. File names in the Parallel I/O File System may be one to EFS_PATH_MAX characters long.

`-eflags eflag` The external file system create and open flags. This argument defaults to O_RDONLY for IMPORT, and O_WRONLY and O_CREAT for EXPORT. Accepted values are O_CREAT, O_EXCL, O_NSHARE, O_RSHARE, O_DEFER, O_NOCTTY, O_TRUNC, O_SYNC, O_APPEND, O_DELAY, O_NDELAY, and O_NONBLOCK. Refer to the AIX open function for a description of these flags.

`-eperms eperm`

The external file system permissions. This argument is only used if the external file must be created. 0640 (-rw-r-----) is the default permission for EXPORT requests.

-pname *piofs_file*

The full path name of the Parallel I/O File System file which is the target of this import or export. File names in the Parallel I/O File System may be one to PIOFS_PATH_MAX characters long.

-pflags *pflag*

The Parallel I/O File System create and open flags. This argument defaults to O_RDONLY for IMPORT, and O_WRONLY and O_CREAT for EXPORT. Accepted values are O_CREAT, O_EXCL, O_NSHARE, O_RSHARE, O_DEFER, O_NOCTTY, O_TRUNC, O_SYNC, O_APPEND, O_DELAY, O_NDELAY, and O_NONBLOCK. Refer to the IBM Parallel I/O File System open function for a description of these flags.

-pperm *pperm*

The Parallel I/O File System file permissions. This argument is only used if the Parallel I/O File System file must be created. 0640 (-rw-r-----) is the default permission for IMPORT requests.

-precsize *recsize*

The Parallel I/O File System file record size. This argument is used only if the Parallel I/O File System file must be created. Record size defaults to the Parallel I/O File System default value.

-pcells *cells*

The number of cells to use if a Parallel I/O File System file must be created. This argument defaults to the Parallel I/O File System default value.

-vbs *vbsn*

Number of records in a vertical group. This argument defaults to 1. Each of the numbers *vbsn*, *vnn*, *hbsn*, and *hnn* may be between 1 and 4,294,967,295, i.e. a 32 bit unsigned integer. Refer to the Installing, Managing, and Using the IBM AIX Parallel I/O File System document for a description of the vbs, vn, hbs, and hn parameters.

-vn *vnn*

Number of vertically interleaved groups. This argument defaults to one. See the -vbs option.

-hbs *hbsn*

Number of physical partitions in a horizontal group. This argument defaults to one. See the -vbs option.

-hn *hnn*

Number of horizontally interleaved groups. This argument defaults to one. See the -vbs option

-subfile *subfilen*

The subfile number (between 0 and *hnn* x *vnn* - 1) of the Parallel I/O File System file that is the target of the import or export. This argument defaults to zero.

-chkptflag {CHECKPOINT | ACTIVE}

Either CHECKPOINT or ACTIVE. This argument applies only to EXPORT and is ignored for IMPORT. If CHECKPOINT is set then the checkpointed version of the file is exported. If flag is set to CHECKPOINT and no checkpoint version of the file exists, an error occurs. The default is ACTIVE.

-list *src_offset dest_offset requested_len*

Each item on the list is an ordered triple of the *src_offset*, *dest_offset*, and *requested_len*, all in bytes. The offsets and length must be unsigned decimal numbers between 0 and 18,446,744,073,609,551,615. If -list is not specified, the entire file is transferred.

-pfile *parm_file*

All of the above parameters override the setting of the same parameters in an optional parameter file whose name may follow the -pfile tag. One line of the parameter file should contain one tag followed by the tag value. The # symbol denotes comments. Refer to [section 4.1.2](#) (page 52) for a description of the *parm_file* format.

-hints *cos_id* Hints to the external file system. For HPSS this is the Class of Service (COS) identifier. A COS ID may be specified to indicate stripe width. For example, if the COS ID for 4-way 3494 tape striping were 4, *cos_id* would be set to "4".

ERROR CONDITIONS.

"A duplicate field has been entered"

"Invalid argument of field value"

"Field size is either too big or inconsistent"

"Insufficient or invalid import/export options specified"

"Open failed (*error text*)"

"Write failed (*error text*)"

"Close failed (*error text*)"

"piofsie entry in piofsie_config_file (efs = *efs_name*) is invalid or not found"

"rsh to invoke piofsie_hpss failed"

"Allocation of pfile buffer failed"

SEE ALSO.

none.

NOTES.

none.

EXAMPLES.

1. Import a file to PIOFS from HPSS.

```
piofsie -efs hpss -op IMPORT -ename /mydir/myfile
```

```
-pname /piofs/bob/myfile
```

2. Import the third gigabyte of HPSS file /mydir/myfile into offset 0 of PIOFS file /piofs/ bob/myfile. Also, import the fifth gigabyte of the HPSS file into offset 2,147,483,648 of the PIOFS file. The record size is set to 1 MB, and the PIOFS file permissions are set to 0644.

```
piofsie -efs hpss -op IMPORT -ename /mydir/myfile  
  
-pname /piofs/bob/myfile -perms 0664 -precsiz 1048576  
  
-list 2147483648 0 1073741824  
  
-list 4294967296 2147483648 1073741824
```

3. Export the third gigabyte of PIOFS file /piofs/bob/myfile into offset 0 of HPSS file / mydir/myfile. Also, export the fifth gigabyte of the PIOFS file into offset 2,147,483,648 of the HPSS file. The record size is set to 1MB, and the HPSS file permissions are set to 0644. Vertical and horizontal interleaves are specified. An HPSS Class of Service identifier is specified. In this example, the Class of Service identifier might indicate 4-way striping to 3490 media.

```
piofsie -op EXPORT -efs hpss -ename /mydir/myfile  
  
-pname /piofs/bob/myfile -perms 0664 -vbs 4 -vn 2  
  
-hbs 3 -hn 3 -list 2147483648 0 1073741824  
  
-list 4294967296 2147483648 1073741824 -hints 4
```

4.1.2. Import/Export Parameter File

The name of an import/export parameter file may optionally be specified as an argument to the piofsie command. The file is used to specify input options for an import or export request. Each line of the parameter file should contain one tag followed by the tag value. Everything following a # symbol denotes comments. The tag names are the same as the parameters to the piofsie command, i.e. -op, -efs, -ename, -eflags, -eperms, -pname, -pflags, -perms, -precsiz, -pcells, -vbs, -vn, -hbs, -hn, -subfile, -chkptflag, -list, -pfile, and -hints. Refer to [section 4.1.1](#) (page 48) for a description of these tags.

Options for the import/export are taken from the input piofsie arguments and the parameter file. Command line arguments take precedence over corresponding values from the parameter file.

The cumulative set of options from the piofsie arguments and parameter file are stored in a temporary parameter file. This file will be a different file from the file named by the -pfile field, but will be created in the directory name associated with the -pfile name if specified. The process id of the issuing program is appended to the -pfile name. If no -pfile name is specified, the file will be generated in the current working directory. In either case, the directory for the generated file must be accessible from the node of the external file system program (e.g. NFS mounted).

The name of the temporary parameter file is RSHed from the client node to the external file system program defined in the /etc/piofs.piofsconfig file (piofsie_hpss). The generated import/export file will be automatically deleted once the request is processed.

The following examples illustrate the formats of sample parameter files.

Example 1:

Import a file to PIOFS from HPSS.

```
# Example 1 parameter file
-op IMPORT # The entire file will be imported.
-efs hpss
-ename /mydir/myfile
-pname /piofs/bob/myfile
```

Example 2:

Import the third gigabyte of HPSS file /mydir/myfile into offset 0 of PIOFS file /piofs/bob/myfile. Also, import the fifth gigabyte of the HPSS file into offset 2,147,483,648 of the PIOFS file. The record size is set to 1 MB.

```
# Example 2 parameter file
-op IMPORT
-efs hpss
-ename /mydir/myfile
-pname /piofs/bob/myfile
-ppperms 0664
-precsize 1048576
-list 2147483648 0 1073741824
-list 4294967296 2147483648 1073741824
```

Example 3:

Export the third gigabyte of PIOFS file /piofs/bob/myfile into offset 0 of HPSS file /mydir/myfile. Also, export the fifth gigabyte of the PIOFS file into offset 2,147,483,648 of the HPSS file. The record size is set to 1 MB, and the HPSS file permissions are set to 0644. Vertical and horizontal interleaves are specified. An HPSS Class of Service identifier is specified. In this example, the Class of Service identifier might indicate 4-way striping to 3490 media.

```
# Example 3 parameter file
-op EXPORT
-efs hpss
-ename mydir/myfile
-pname /piofs/bob/myfile
-eperms 0664
-vbs 4
-vn 2
-hbs 3
-hn 3
-list 2147483648 0 4294967296 1073741824
-list 4294967296 2147483648 1073741824
-hints 4 # Hints are
# used as a means to specify HPSS striping
```

4.1.3. Configuration Variables and Setup

For information on how to configure HPSS for PIOFS Import/Export support, refer to the HPSS System Administration Guide. The format of the Parallel I/O File System Import/Export configuration file is also described in the following paragraphs.

The HPSS PIOFS (Parallel I/O File System) Import/Export program must be executed on an IBM SP that has access to PIOFS. The program is executed as an on-demand process using the piofsie command, or by an application calling the piofsiep API. Environment variables may be set to override the default values by modifying the PIOFS Import/Export configuration file. Refer to "Optional Entries" in the paragraphs below.

Configuration variables for the PIOFS Import/Export program are set in the PIOFS Import/Export configuration file accessible on each client node. The name of the import/export configuration file is

`/etc/piofs.piofscfg.ie.`

The import/export entries which may be defined in the PIOFS Import/Export configuration file are as follows:

1. REQUIRED ENTRY.

The entry which defines the external file system(s) which support PIOFS import and export operations is a required entry.

`-efs hpss efs_internet_address efs_program_name`

shows the format of the required entry, where

efs_internet_address is the IP address of the node where the import/export program executes.

efs_program_name is the path name of the import/export program.

For example:

```
efs hpss 129.40.33.15 /usr/lpp/hpss/bin/piofsie_hpss
```

2. OPTIONAL ENTRIES.

The following entries are optional, and must be specified only if system defaults for the Name Server DCE name, Bitfile Server DCE name, or client keytab path name are not being used. If defaults are not being used, entries with the following formats must be specified:

HPSS_HPNS_NAME followed by the Name Server name

HPSS_BFS_NAME followed by the Bitfile Server name

HPSS_KTAB_PATH followed by the client keytab path name

For example,

```
HPSS_HPNS_NAME /./hpss/cns  
HPSS_BFS_NAME /./hpss/bfs  
HPSS_KTAB_PATH /krb5/hpssclient.keytab
```

5. User Utilities

[WARNING: at LC, these HPSS utilities are currently for storage system administrators only, NOT for general users (and no LC man pages are provided). See [EZSTORAGE](http://www.llnl.gov/LCdocs/ezstorage) (URL: <http://www.llnl.gov/LCdocs/ezstorage>) for instructions on three other storage tools that *are* available for general users, including LSTORAGE, CHMODSTG, and CHGRPSTG.]

HPSS provides a set of utilities for administrators and users. The majority of the HPSS utilities are for administrators, and are defined in the HPSS Administration Guide. Those utilities applicable to users are documented in this chapter. There are also man pages for these utilities. Note: The user must have a DCE login prior to using the utilities.

5.1. Utilities

The user utilities defined in the chapter are:

- chacl
- lsacl
- lshpss

5.1.1. CHACL (Change ACL utility)

NOTE: CHACL is currently for LC storage system administrators only, NOT for general users.

SYNOPSIS.

```
chacl [-r -u -b] < file
```

DESCRIPTION.

The chacl utility is used to change discretionary access control information associated with a specified object. The output from lsacl is in the correct format for input to chacl. The input to chacl for an ACL consists of two sections: the comments section and the ACL entries section. The comments section contains three lines: name of the object, the object owner, and the object owning group. The ACL entries section contains lines with three fields separated by a colon. The first field contains the ACL entry tag type. The second field contains the ACL entry qualifier. The third field contains the access permissions.

OUTPUT FORMAT.

```
#file:<filename>
#owner:<uid>
#group:<gid>
user::<perm>
mask::<perm>
user:<uid>:<perm>
user:<uid2>:<perm>
group::<perm>
group:<gid>:<perm>
group:<gid2>:<perm>
other::<perm>
```

The types of permissions are:

r - read
w - write
x - execute
i - insert
d - delete
a - append
m - modify

The utility `chacl` is initiated from the command line. The utility differs from some other HPSS utilities in that it uses the Client API library. The user must have issued a `dce_login`. If HPSS were configured without the default values, two environment variables must be set. The environment variables are the CDS names of the Name Server and the Bitfile Server: `HPSS_HPNS_NAME` and `HPSS_BFS_NAME`.

PARAMETERS.

`-r` removes the specified ACL entries. Base entries are not affected.

`-u` updates the specified ACL entries. The entries are added if they do not already exist.

`-b` removes all entries except the three base entries, `user::`, `group::`, and `other::`.

file specifies the file containing the ACL information to change (usually output from `LSACL`).

SEE ALSO.

`lsacl`

NOTES.

none.

EXAMPLES.

The following steps add the user with ID 7004 with read permission to the ACL list of the file `/ampex/test.txt`.

```
dce_login hpss_client1

export HPSS_HPNS_NAME=././hpss/cns
export HPSS_BFS_NAME=././hpss/bfs

lsacl -f /ampex/test.txt > acl_format

cat acl_format
#file: /ampex/test.txt
#owner:1006
#group:system
user::rwx---m
mask::r-x---
group::r-x---
```

Now edit `acl_format` to add the line `"user:7004:r-----"`

```
cat acl_format
#file: /ampex/test.txt
#owner:1006
#group:system
user::rwx---m
user:7004:r-----
mask::r-x---
group::r-x---

chacl -u < acl_format
```

5.1.2. LSACL (List ACL utility)

NOTE: LSACL is currently for LC storage system administrators only, NOT for general users.

SYNOPSIS.

```
lsacl [-f < object_path_name ]
```

DESCRIPTION.

The lsacl utility is used to list access control information associated with a specific path name. The output from lsacl is in the correct format for input to chacl (change ACL). The output from lsacl for an object ACL consists of two sections: the comments section and the ACL entries section. The comments section contains three lines: name of the object, the object owner, and the object owning group. The ACL entries section contains lines with three fields separated by a colon. The first field contains the ACL entry tag type. The second field contains the ACL entry qualifier. The third field contains the access permissions.

OUTPUT FORMAT.

```
#file:<filename>
#owner:<uid>
#group:<gid>
user::<perm>
mask::<perm>
user:<uid>:<perm>
user:<uid2>:<perm>
group::<perm>
group:<gid>:<perm>
group:<gid2>:<perm>
other::<perm>
```

The types of permissions are:

```
r - read
w - write
x - execute
i - insert
d - delete
a - append
m - modify
```

The utility lsacl is initiated from the command line. The pathname is the key input value. The utility differs from some other HPSS utilities in that it uses the Client API library. The user must have issued a dce_login. If HPSS were configured without the default values, two environment variables must be set. The environment variables are the CDS names of the Name Server and the Bitfile Server: HPSS_HPNS_NAME and HPSS_BFS_NAME.

PARAMETERS.

-f specifies the HPSS pathname of the object whose ACLs are to be listed.
object_path_name

SEE ALSO.

chacl

NOTES.

none.

EXAMPLES.

The following steps list the ACLs associated with the directory /ampex owned by the HPSS user hpss_client1.

```
dce_login hpss_client1
```

```
export HPSS_HPNS_NAME=././hpss/cns
export HPSS_BFS_NAME=././hpss/bfs
```

```
lsacl -f /ampex
#file: /ampex
#owner:1006
#group:system
user::rwx---m
mask::r-x---
group::r-x---
```

5.1.3. LSHPSS (List information about HPSS)

NOTE: LSHPSS is currently for LC storage system administrators only, NOT for general users.

SYNOPSIS.

```
lshpss [ options ]
```

DESCRIPTION.

The lshpss utility displays HPSS resources, such as Class of Service, Hierarchy, and Storage Class. Before running this script, you must be authorized to access the SFS files. This is accomplished by entering dce_login for an authorized DCE ID.

PARAMETERS.

- | | |
|----------------------|---|
| -s <i>sfs_server</i> | specifies optional Encina SFS Server. The SFS Server can also be set by setting the ENCINA_SFS_SERVER environment variable (e.g., using export ENCINA_SFS_SERVER=./encina/sfs/hpss). |
| -cos | shows the class of service list. |
| -hier | shows hierarchy list. |
| -sc | shows storage class list. |
| -vol | shows physical volumes. |
| -dev | shows devices and drives. |
| -svr | shows HPSS servers. |
| -mvr | shows HPSS movers. |
| -h | shows this help message. |

SEE ALSO.

none.

NOTES.

The option of interest to most users is the -cos option. This option allows the user to view a list of all defined Classes of Service.

EXAMPLES.

The following steps list the classes of service, hierarchies, and storage classes.

```
dce_login hpss_client1
```

```
lshpss -cos -hier -sc
```

Sample output from this example is shown below.

| - Class of Service List - | | | | | | | | | |
|---------------------------|-----|-----------------------|-------------------|-------|--------|------|----|-----|--|
| CID | HID | Description | OptimumAccessSize | MinFS | MaxFS | Rate | AL | STG | |
| 1 | 1 | 1-w SSA -> 1-w 3490E | 4MB | 0 | 32000M | 4096 | 0 | Opn | |
| 5 | 5 | 1-w 3590 | 4MB | 0 | 32000M | 4096 | 0 | Opn | |
| 10 | 10 | 3-w SCSI -> 2-w 3490E | 4MB | 0 | 32000M | 4096 | 0 | Opn | |

CID = COS ID | HID = Hierarchy ID | MaxFS = MaxFileSize
 Rate = TransferRate (Unit is KBytes) | AL = AvgLatency
 Stg = StageCode (NoSt="No Stage" Opn="On Open" OpnA="On Open Async")

| - Hierarchy List - | | | |
|--------------------|-----------------------|--------|----------------|
| ID | Description | Levels | StorageClassID |
| 1 | 1-w SSA -> 1-w 3490E | 2 | 1 --> 10 |
| 5 | 1-w 3590 | 1 | 19 |
| 10 | 3-w SCSI -> 2-w 3490E | 2 | 7 --> 16 |

ID = Hierarchy ID

| - Storage Class List - | | | | | | | | | | | | | |
|------------------------|--------------------------|-----------|-------|--------|---|-------|------|-----|-----|----|----|--------|----|
| ID | Description | TransRate | EstSz | StpLen | W | VVBlk | MBlk | Crt | Wrn | MP | PP | OASize | SS |
| 1 | 1-way SSA Disk (H1-L1) | 0 | 512MB | 64k | 1 | 4k | 4k | 95 | 80 | 1 | 1 | 0 | |
| 5 | 1-way SCSI Disk (H8-L1) | 0 | 1GB | 256k | 1 | 256k | 4k | 95 | 80 | 6 | 6 | 0 | |
| 10 | 1-way 3490E Tape (H1-L2) | 0 | 800MB | 1MB | 1 | 1MB | 32k | 95 | 80 | 0 | 0 | 0 | |

| | | | | | |
|--------|-------------------|--------|---------------------|-----|--------------------|
| VVBlk | =VVBLOCKSize | MBlk | =MediaBLOCKSize | Crt | =CriticalThreshold |
| Wrn | =WarningThreshold | MP | =MigrationPolicyID | PP | =PurgePolicyID |
| OASize | =OptimAccessSize | SSegSz | =StorageSegmentSize | | |

6. Storage Groups

A group is a named set of users created by a system administrator to enable easier file sharing among group members. On a production machine, assigning a file to a group (with CHGRP) enables other group members to take advantage of whatever group permissions you have declared (with CHMOD) to list, read, or overwrite the file.

Groups on the HPSS storage systems (open and secure) at LC, however, are managed independently of groups on LC production machines. This means that a file's group assignment may change when you store or retrieve it, and that you must use special tools to monitor your group membership(s) on the storage machines. This section supplements the general advice about "[Using Groups](http://www.llnl.gov/LCdocs/ezfiles/index.jsp?show=s9) (URL: <http://www.llnl.gov/LCdocs/ezfiles/index.jsp?show=s9>)" in the EZFILES guide with specific instructions for storage groups.

YOUR STORAGE GROUPS:

LC uses standard UNIX tools (such as GROUPS) and files (such as /etc/group) to manage its nonstorage groups. But LC uses DCE (Distributed Computing Environment) tools to manage its storage groups. Consequently, the group(s) to which you belong on the storage systems (open and secure separately), and to which you can assign your stored files, may differ from those on some or all LC production machines (even machines with DCE-managed passwords).

You can discover your storage group names by running the DCE RGY_EDIT tool, but it is easier to use the DCECP utility (which also reveals group member names, as shown below). This dialog shows how to run DCECP (Distributed Computing Environment Control Program) on any LC DEC, Sun, or IBM machine to reveal which HPSS storage groups you belong to (where *uname* is your user name):

```
User: dcecp
R/Us: dcecp> user show uname
Rtne: {fullname {Leonora Florestan}}
      [four other data lines...]
      {groups group1, group2, group3...}
      [many other data lines...]
R/Us: dcecp> quit
```

In this report, *group1* is always the single-member group with the same name as your login name (and on open HPSS it is often your only storage group).

STORAGE GROUP MEMBERSHIP:

This dialog shows how to run DCECP on any LC DEC, Sun, or IBM machine to reveal all the storage-user members of a specified HPSS storage group (here *gname*):

```
User: dcecp
R/Us: dcecp> group list gname
Rtne: /.../server/user1
      /.../server/user2
      /.../server/user3
      [other data lines...]
R/Us: dcecp> quit
```

Group members are listed one per line, each preceded by the domain name of its DCE registry server.

WHEN YOU STORE OR RETRIEVE:

If you assign a file to a (nondefault) group and later store the file in HPSS, that group assignment is lost. HPSS always associates the stored file with your single-member username group by default. If you assign two files each to a different group and later store both files, their different group assignments are both lost. Both stored files revert to the same username group; neither keeps either nondefault group you assigned. Likewise, if you assign a stored file to a (nondefault) group and later retrieve it, that storage-group assignment is lost as well.

This loss of group assignment when storing or retrieving files occurs not only if your storage groups differ from your online groups, as you might expect, but also even if you belong to the same group on a production machine and on storage. Currently, (nondefault) group assignment simply does not persist during FTP-mediated file transfers among LC machines, including transfers to or from storage.

CHANGING STORAGE GROUPS:

To enable or limit the sharing of stored files, you can change the group assignment of a file *after* you store it either by using the supplementary FTP command

```
quote site chgrp grpname filename
```

for example

```
quote site chgrp us_cit myfile3
```

or by using the NFT command

```
chgrp grpname filename
```

just as you would use the CHGRP utility on a production machine. (Despite its name, NFT's GROUP command begins asynchronous file transfers and has nothing to do with managing the file-permission groups of stored files.) You can assign a stored file only to a storage group that you belong to. Similarly, the supplementary FTP command

```
quote site chmod mode filename
```

will set a stored file's (group) permissions to the octal *mode* that you specify.

NOTE: on LC's Compaq and Linux machines, you can also change a stored file's group by using the special CHGRPSTG tool and change its permissions by using the CHMODSTG tool. Instructions for and examples of these local tools to manage storage groups appear in the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide.

RELEVANT FORMS:

To create, delete, or change the membership of a group on either the open or SCF storage machines, you must submit one of these (signed) forms to the LC Hotline (L-63).

SCF-2 Create/Update Group
SCF-9 Delete Group

The blank forms are available by request from the Hotline (page 00) and from these LC web sites:

Open: <http://www.llnl.gov/lcforms/forms.html>
SCF: <http://www.scf.cln/lcforms>

Appendix A. Acronyms

| | |
|-------|---|
| ACL | Access Control List |
| ACLS | Automated Cartridge System Library Software (Science Tech. Corp.) |
| AIX | Advanced Interactive Executive |
| API | Application Program Interface |
| CDS | Cell Directory Server |
| COS | Class of Service |
| DCE | Distributed Computing Environment |
| EFS | External File System |
| FTP | File Transfer Protocol |
| gid | Group Identifier |
| GSS | Generic Security Service |
| HIPPI | High Performance Parallel Interface |
| HPSS | High Performance Storage System |
| IBM | International Business Machines Corporation |
| LaRC | Langley Research Center |
| LANL | Los Alamos National Laboratory |
| LLNL | Lawrence Livermore National Laboratory |
| NASA | National Aeronautics and Space Administration |
| ORNL | Oak Ridge National Laboratory |
| ID | Identifier |
| IEEE | Institute of Electrical and Electronics Engineers |
| I/O | Input/Output |
| IP | Internet Protocol |
| IPI | Intelligent Peripheral Interface |
| NFS | Network File System |
| OSF | Open Software Foundation |
| PFTP | Parallel File Transfer Protocol |
| PIOFS | Parallel I/O File System |
| RISC | Reduced Instruction Set Computer |
| SFS | Structured File Server |
| SNL | Sandia National Laboratories |
| SP | Scalable Processor |
| TCP | Transmission Control Protocol |
| uid | User Identifier |
| VV | Virtual Volume |

Appendix B. References

1. File Transfer Protocol, RFC-0959, October 1985.
2. HPSS Error Messages Manual, August 1997.
3. HPSS Programmer's Guide Reference, Volume 1, August 1997.
4. HPSS Programmer's Guide Reference, Volume 2, August 1997.
5. HPSS System Administration Guide, August 1997.
6. Installing, Managing, and Using the IBM AIX Parallel I/O File System, Document Number H34-6065-00.
7. Network File System Specification, RFC-1094, DDN Network Information Center, SRI International, Menlo Park, Ca.
8. OSF DCE User's Guide and Reference, Prentice Hall, Englewood Cliffs, N. J.

Appendix C. HPSS/NSL UniTree Differences

C.1. Trash Cans Discontinued

Under NSL UniTree, when you "deleted" a stored file, it moved to a child of your home (storage) directory called .trash (under an altered name), where it lingered for a predefined period before its final, irreversible removal from the storage system. Under HPSS, the delayed deletion of stored files using .trash, and the **gettrash** and **settrash** control commands, have been discontinued. Under HPSS, you cannot recover a file once you have deleted it from storage.

C.2. Supplementary Commands Changed

The acceptable nonstandard FTP commands (those you must prefix with **quote**, **site**, or both) differ significantly between HPSS and NSL UniTree. The chart below compares the two sets and indicates that even when a command has the same name, its syntax (and hence its specific role) has sometimes changed. See the [Site Commands](#) (page 23) section for usage details on the HPSS-supported supplementary commands.

| Supplementary FTP Commands | |
|----------------------------|-------------------------------|
| HPSS (current) | NSL UniTree (discontinued) |
| quote site chgrp | quote chgrp |
| quote site chgid | |
| | quote chhid |
| quote site chmod | quote chmod |
| quote site chown | quote chown |
| | quote chuacct |
| quote site chuid | |
| | quote direct |
| | quote display |
| | quote getfam |
| | quote gethid |
| | quote getndup |
| | quote gettrash |
| | quote getuacct |
| | quote link |
| | quote listopt |
| quote site setcos | |
| | quote setfam |
| | quote sethid |
| | quote setndup |
| | quote settrash |
| | quote setuacct |
| quote site stage | quote stage [syntax changed] |
| | quote symlink |
| | quote umask |
| quote site wait | quote wait [syntax changed] |
| | |
| | site groups |
| | site idle |

C.3. Listing Options Changed

Under HPSS, the FTP server daemon no longer supports any of the following former list-control options (once used with FTP's or NFT's **dir**, for example):

```
-C -L -Q -R -X (all uppercase)
-f -t      -r -x (all lowercase)
```

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

(C) Copyright 2003 The Regents of the University of California. All rights reserved.

Keyword Index

To see an alphabetical list of keywords for this document, consult the [next section](#) (page 74).

| Keyword | Description |
|-----------------------------|---|
| <u>entire</u> | This entire document. |
| <u>title</u> | The name of this document. |
| <u>scope</u> | Topics covered in HPSS Guide. |
| <u>availability</u> | Where these programs run. |
| <u>who</u> | Who to contact for assistance. |
| <u>introduction</u> | Role and goals of HPSS Guide. |
| <u>organization</u> | Chapter contents summarized. |
| <u>style</u> | Typographic conventions used. |
| <u>ezstorage</u> | Storage tasks and limits summarized. |
| <u>overview</u> | HPSS concepts, requirements, issues. |
| <u>interfaces</u> | Alternative HPSS user interfaces. |
| <u>ftp-summary</u> | FTP as storage interface, intro. |
| <u>pftp-summary</u> | Parallel FTP as storage interface, intro. |
| <u>nfs-summary</u> | NFS as storage interface, intro. |
| <u>piofs-summary</u> | PIOFS as storage interface, intro. |
| <u>nft-summary</u> | NFT as storage interface, intro (local). |
| <u>utility-summary</u> | Storage "utility" commands summarized. |
| <u>storage-concepts</u> | Three concepts used by HPSS. |
| <u>class-of-service</u> | "Class of service" (COS) defined. |
| <u>storage-class</u> | Storage classes to group storage media. |
| <u>storage-hierarchy</u> | Stage and migrate between storage levels. |
| <u>interface-comparison</u> | Comparison of storage interfaces. |
| <u>user-id</u> | Administering storage IDs. |
| <u>dce-role</u> | DCE prerequisite for storage utilities. |
| <u>ftp</u> | FTP storage options, details. |
| <u>site-commands</u> | Extra SITE-prefix HPSS/FTP options. |
| <u>setcos</u> | Specify class of service. |
| <u>chgid</u> | Change group by using ID. |
| <u>chgrp</u> | Change group by using name. |
| <u>chmod</u> | Change access permissions. |
| <u>chown</u> | Change stored file's owner. |
| <u>chuid</u> | Change owner by using ID. |
| <u>stage</u> | Prepare migrated file for retrieval. |
| <u>wait</u> | Set WAIT option for migrated files. |
| <u>list-storage</u> | Special stored-file reporting option. |
| <u>pftp</u> | Parallel FTP (PFTP) storage options. |
| <u>pftp-options</u> | PFTP storage options. |
| <u>pappend</u> | Parallel file append. |
| <u>pput</u> | Parallel file store. |
| <u>mpput</u> | Parallel multiple file store. |
| <u>pget</u> | Parallel file retrieve. |
| <u>mpget</u> | Parallel multiple file retrieve. |
| <u>psocket</u> | Use parallel socket transfer. |
| <u>pipi3</u> | Use parallel IPI-3 transfers. |
| <u>setpwidth</u> | Set transfer stripe width. |
| <u>setpblocksize</u> | Set transfer block size. |

| | |
|---|--|
| <u>parallel</u> | Toggle parallel transfers (LLNL only). |
| <u>htar</u> | HTAR front end to PFTP (LLNL only). |
| <u>piofs</u> | PIOFS import/export details. |
| <u>piofs-options</u> | PIOFS storage options. |
| <u>piofsie</u> | Import or export a PIOFS file. |
| <u>piofsie-parameters</u> | Parameter file for PIOFS import/export. |
| <u>piofsie-configuration</u> | Setup of PIOFS config. variables. |
| <u>user-utilities</u> | Storage control tools. |
| <u>storage-tools</u> | Storage control tools. |
| <u>chacl</u> | Change access control list. |
| <u>lsacl</u> | Display access control list. |
| <u>lshpss</u> | Display HPSS configuration info. |
| <u>storage-groups</u> | Reporting, using, changing storage groups. |
| <u>acronyms</u> | Storage and related acronyms. |
| <u>references</u> | HPSS reference suggestions. |
| <u>differences</u> | HPSS/NSL UniTree diffs summarized. |
| <u>trash-cans</u> | Trash can directories discontinued. |
| <u>supplementary-commands</u> | Changes in quote, site commands. |
| <u>list-changes</u> | Many list options discontinued. |
| <u>index</u> | The structural index of keywords. |
| <u>a</u> | The alphabetical index of keywords. |
| <u>date</u> | The latest changes to HPSS Guide. |
| <u>revisions</u> | The complete revision history. |

Alphabetical List of Keywords

| Keyword ----- | Description ----- |
|------------------------------|---|
| <u>a</u> | The alphabetical index of keywords. |
| <u>acronyms</u> | Storage and related acronyms. |
| <u>availability</u> | Where these programs run. |
| <u>chacl</u> | Change access control list. |
| <u>chgid</u> | Change group by using ID. |
| <u>chgrp</u> | Change group by using name. |
| <u>chmod</u> | Change access permissions. |
| <u>chown</u> | Change stored file's owner. |
| <u>chuid</u> | Change owner by using ID. |
| <u>class-of-service</u> | "Class of service" (COS) defined. |
| <u>date</u> | The latest changes to HPSS Guide. |
| <u>dce-role</u> | DCE prerequisite for storage utilities. |
| <u>differences</u> | HPSS/NSL UniTree diffs summarized. |
| <u>entire</u> | This entire document. |
| <u>ezstorage</u> | Storage tasks and limits summarized. |
| <u>ftp</u> | FTP storage options, details. |
| <u>ftp-summary</u> | FTP as storage interface, intro. |
| <u>htar</u> | HTAR front end to PFTP (LLNL only). |
| <u>index</u> | The structural index of keywords. |
| <u>interface-comparison</u> | Comparison of storage interfaces. |
| <u>interfaces</u> | Alternative HPSS user interfaces. |
| <u>introduction</u> | Role and goals of HPSS Guide. |
| <u>list-changes</u> | Many list options discontinued. |
| <u>list-storage</u> | Special stored-file reporting option. |
| <u>lsacl</u> | Display access control list. |
| <u>lshpss</u> | Display HPSS configuration info. |
| <u>mpget</u> | Parallel multiple file retrieve. |
| <u>mpput</u> | Parallel multiple file store. |
| <u>nfs-summary</u> | NFS as storage interface, intro. |
| <u>nft-summary</u> | NFT as storage interface, intro (local). |
| <u>organization</u> | Chapter contents summarized. |
| <u>overview</u> | HPSS concepts, requirements, issues. |
| <u>pappend</u> | Parallel file append. |
| <u>parallel</u> | Toggle parallel transfers (LLNL only). |
| <u>pftp</u> | Parallel FTP (PFTP) storage options. |
| <u>pftp-options</u> | PFTP storage options. |
| <u>pftp-summary</u> | Parallel FTP as storage interface, intro. |
| <u>pget</u> | Parallel file retrieve. |
| <u>piofs</u> | PIOFS import/export details. |
| <u>piofs-options</u> | PIOFS storage options. |
| <u>piofs-summary</u> | PIOFS as storage interface, intro. |
| <u>piofsie</u> | Import or export a PIOFS file. |
| <u>piofsie-configuration</u> | Setup of PIOFS config. variables. |
| <u>piofsie-parameters</u> | Parameter file for PIOFS import/export. |
| <u>pipi3</u> | Use parallel IPI-3 transfers. |
| <u>pput</u> | Parallel file store. |
| <u>psocket</u> | Use parallel socket transfer. |
| <u>references</u> | HPSS reference suggestions. |
| <u>revisions</u> | The complete revision history. |
| <u>scope</u> | Topics covered in HPSS Guide. |
| <u>setcos</u> | Specify class of service. |

| | |
|-------------------------------|--|
| <u>setpblocksize</u> | Set transfer block size. |
| <u>setpwidth</u> | Set transfer stripe width. |
| <u>site-commands</u> | Extra SITE-prefix HPSS/FTP options. |
| <u>stage</u> | Prepare migrated file for retrieval. |
| <u>storage-class</u> | Storage classes to group storage media. |
| <u>storage-concepts</u> | Three concepts used by HPSS. |
| <u>storage-groups</u> | Reporting, using, changing storage groups. |
| <u>storage-hierarchy</u> | Stage and migrate between storage levels. |
| <u>storage-tools</u> | Storage control tools. |
| <u>style</u> | Typographic conventions used. |
| <u>supplementary-commands</u> | Changes in quote, site commands. |
| <u>title</u> | The name of this document. |
| <u>trash-cans</u> | Trash can directories discontinued. |
| <u>user-id</u> | Administering storage IDs. |
| <u>user-utilities</u> | Storage control tools. |
| <u>utility-summary</u> | Storage "utility" commands summarized. |
| <u>wait</u> | Set WAIT option for migrated files. |
| <u>who</u> | Who to contact for assistance. |

Date and Revisions

| Revision Date ----- | Keyword Affected ----- | Description of Change ----- |
|---------------------------|--|---|
| 02Dec03 | <u>ezstorage</u> <u>htar</u> | Size and scope details updated. Limits updated, -F option noted. |
| 05May03 | <u>parallel</u> | More automatic parallel transfers. |
| 18Feb03 | <u>introduction</u> <u>ezstorage</u> <u>ftp-summary</u> <u>ftp</u> | SFTP not an HPSS interface. SFTP not an HPSS interface. SFTP uses nonFTP daemon. SFTP uses nonFTP daemon. |
| 12Nov02 | <u>class-of-service</u> <u>interface-comparison</u> <u>setcos</u> | NETMON tracking role noted. NETMON tracking role noted. NETMON uses HPSS COSs to track FTP. |
| 18Jun02 | <u>class-of-service</u> <u>setcos</u> <u>ftp-summary</u> <u>ftp</u> <u>htar</u> <u>ezstorage</u> | Details and roles updated. COS table revised, expanded to include duplicate copy info. MDELETE warning added. MDELETE warning added. Details revised, Y added. Maximum file size clarified. |
| 23Aug01 | <u>pftp</u> <u>htar</u> <u>introduction</u> <u>ezstorage</u> <u>index</u> | HTAR as PFTP front end. Brief feature summary added. HTAR storage role noted. HTAR storage role noted. New keyword for new section. |
| 05Jul01 | <u>pftp</u> <u>parallel</u> <u>interfaces</u> <u>ftp</u> <u>introduction</u> <u>index</u> | Now default, many local usage details added for all commands. Local command explained (added). FTP/PFTP comparison updated. Actually invokes PFTP now. FTP/PFTP interaction noted. New keyword for new section. |
| 23May00 | <u>scope</u> <u>introduction</u> <u>ezstorage</u> <u>chgrp</u> <u>chmod</u> <u>list-storage</u> <u>user-utilities</u> <u>storage-groups</u> | EZSTORAGE guide noted. EZSTORAGE guide noted. Cross reference added. CHGRPSTG tool in EZSTORAGE noted. CHMODSTG tool in EZSTORAGE noted. LSTORAGE tool in EZSTORAGE noted. LSTORAGE, CHMODSTG, CHGRPSTG noted. New group-relevant tools noted. |
| 02Apr99 | <u>ezstorage</u> <u>ftp</u> | Firewall now blocks outside access. Firewall now blocks outside access. |

| | | |
|---------|---|---|
| 08Feb99 | <u>introduction</u> <u>ezstorage</u> <u>ftp</u> | No SCP access to storage. No SCP access to storage. Firewall alert added. |
| 10Sep98 | <u>storage-groups</u> | New section on using storage groups. |
| 13Jul98 | <u>ezstorage</u> <u>nft-summary</u> <u>differences</u> <u>index</u> <u>interfaces</u> <u>site-commands</u> <u>setcos</u> <u>user-utilities</u> | New section summarizes storage. New section introduces NFT. New section contrasts HPSS and UniTree. New keywords for new sections. Warnings when not at LC. Warnings about quote syntax. Local COSs listed. Warnings that not generally avail. |
| 15Jun98 | <u>entire</u> <u>index</u> <u>a</u> | Cross references added throughout. Keyword index added. Alphabetical index added. |
| 11Jun98 | <u>entire</u> | First edition of LC HPSS Guide. |

TRG (02Dec03)

UCRL-WEB-201755

Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

TRG (02Dec03) Contact on the OCF: lc-hotline@llnl.gov, on the SCF: lc-hotline@pop.llnl.gov